VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ZÁTĚŽOVÉ TESTOVÁNÍ POČÍTAČOVÝCH SÍTÍ

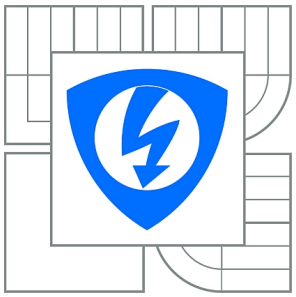
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE DANIEL BOLEK
AUTHOR

BRNO 2014

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

ZÁTĚŽOVÉ TESTOVÁNÍ POČÍTAČOVÝCH SÍTÍ

STRESS TESTING OF COMPUTER NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

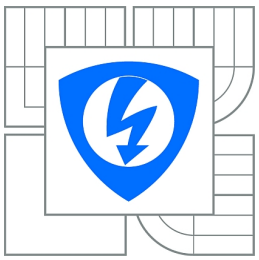
AUTOR PRÁCE — DANIEL BOLEK

AUTHOR

VEDOUCÍ PRÁCE — Ing. JAN HAJNÝ, Ph.D.

SUPERVISOR

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Daniel Bolek	***ID:***	146791
Ročník:	3	***Akademický rok:***	2013/2014

NÁZEV TÉMATU:

Zátěžové testování počítačových sítí

POKYNY PRO VYPRACOVÁNÍ:

Nainstalujte server s OS Linux a službami webserveru, FTP serveru a SSH serveru. Změřte propustnost a odezvu serveru pro různé služby a velikosti datových souborů pomocí Spirent Avalanche. Analyzujte možnosti DDoS útoků poskytované Spirent Avalanche. Analyzujte současný stav v oblasti DDoS útoků, jejich četnosti výskytu a parametry. Změřte vliv DDoS útoků na služby poskytované serverem. Navrhněte ochranné prostředky proti vybraným DDoS útokům.

DOPORUČENÁ LITERATURA:

[1] STALLINGS, William. Cryptography and network security: principles and practice. Seventh edition. xix, 731 pages. ISBN 01-333-5469-5.

[2] Spirent Support [online]. 2013 [cit. 2013-10-07]. Dostupné z: http://support.spirentcom.com

[3] Root.cz [online]. 2013 [cit. 2013-10-07]. Dostupné z: www.root.cz

Termín zadání:	10.2.2014	***Termín odevzdání:***	4.6.2014

Vedoucí práce: Ing. Jan Hajný, Ph.D.
Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Bakalářská práce se zabývá zátěžovým testováním serveru s operačním systémem Linux. Teoretická část se stručně věnuje historií Linuxu, výběru distibuce Debian a serverovým aplikacím Apache a Vsftpd. Následně je popsáno zařízení Spirent Avalanche 3100, sloužící ke generování potřebné zátěže a interpretaci výsledků provedených testů. Práce se dále věnuje problematice DDoS útoků, analýzuje současný stav v této oblasti a popisuje útoky poskytované zařízením Avalanche 3100. Dále ukazuje možné řešení, jak postupovat při návrhu ochranných prostředků proti DDoS útokům.
Praktická část je zaměřena na instalaci operačního systému Linux, implementaci a konfiguraci služeb webového, FTP a SSH serveru a nastavení firewallu. Tento server je poté podroben zátěžovým testům. Cílem je otestovat úspěšnost HTTP a FTP serveru pro různou velikost generované zátěže a zjistit vliv velikosti stahovaného souboru na odezvu serveru. Cílem následující části je změřit vliv vybraných DDoS útoků na služby poskytované serverem. Poslední část je věnována testování ochran proti SynFlood útoku.

KLÍČOVÁ SLOVA

DoS, DDoS, SynFlood, Linux server, Avalanche 3100, zátěžové testování

ABSTRACT

Bachelor's thesis deals with stress testing of server running on operating system Linux. Theoretical part of this project briefly describes the history of Linux, choice of distribution Debian and the server's applications Apache and Vsftpd. Then I describe device Spirent Avalanche 3100, which is designed to generate load and interprets results of tests. Semestral project also deals with DDoS attacks, analyzes current state in this field and describe those DDoS attacks, which provides device Avalanche 3100. Then shows possible solution, how to proceed, if we want to design protection against DDoS attacks. Practical part of semestral project is focused on installation of a operating system Linux, implementation and configuration web, FTP and SSH server services and firewall settings. After that the server is subjected to stress testing. The main aim is to test the success of HTTP and FTP server for different load height and determine, whether size of downloaded file has an effect to response time of the server. The aim of following section is to measure the impact of the choosen DDoS attacks. Protection against SynFlood attack is tested in the last part.

KEYWORDS

DoS, DDoS, SynFlood, Linux server, Avalanche 3100, stress testing

BOLEK, Daniel *Zátěžové testování počítačových sítí*: bakalářská práce. Místo: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 85 s. Vedoucí práce byl Ing. Jan Hajný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Zátěžové testování počítačových sítí“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Místo ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Janu Hajnému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Místo ....

(podpis autora)

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře vybudované v rámci projektu CZ.1.05/2.1.00/03.0072
Centrum senzorických, informačních a komunikačních systémů (SIX)
operačního programu Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Webové a datové služby jsou v dnešní době stále více využívány. Staly se součástí dnešního světa. Ať už jde o vzdělávání, umění, podnikání, či jinou lidskou aktivitu, všude nacházejí své právoplatné místo. Kvůli své popularitě ovšem bývají tyto služby často terčem útoků. Jednoduchým příkladem nám mohou být internetové obchody. V jejich případě znamená odepření webové služby přímou finanční ztrátu. Podobných příkladů bychom mohli najít celou řadu.

Z hlediska předcházení možným hrozbám a omezení jejich následků je dobré znát výkonnostní limity zařízení, na kterých dané služby provozujeme. Tyto limity a další jiná úskalí nám může pomoci odhalit specializované zařízení *Avalanche 3100* od společnosti *Spirent Communications*. Díky tomuto zařízení jsme schopni realizovat celé spektrum zátěžových testů a cílených útoků, získat potřebné výsledky a na jejich základě můžeme navrhnout potřebná opatření ke zvýšeni bezpečnosti a odolnosti naších zařízení.

Tato práce se věnuje zátěžovému testování serveru poskytující webové a datové služby za pomocí specializovaného zařízení *Avalanche 3100*. První kapitola se bude věnovat stručnému seznámení s operačním systémem Linux a jeho distribucí *Debian*. Druhá kapitola ve zkratce představí aplikace *Apache* a *Vsftpd*, které budou poskytovat webové a datové služby. V následující kapitole bude popsána problematika zátěžového testování a blíže se seznámíme se zařízením *Avalanche 3100*. V rámci této kapitoly si také ukážeme jak se zařízením pracovat pomocí aplikací *TestCenter Layer 4-7 Application* a *TestCenter Results Analyzer*. Další kapitola se zabývá hrozbami DDoS útoků, analýzou jejich současného stavu a stručným popisem DDoS útoků, které poskytuje zařízení *Avalanche 3100*. Bude zde také ukázáno obecné řešení, jak postupovat při návrhu ochranných prostředků proti těmto útokům. Topologie laboratorní sítě bude popsáná v kapitole páté.

V šesté kapitole budou nejdříve popsány praktické kroky potřebné k instalaci operačního systému Linux, dále k implementaci a konfiguraci serverových aplikací, firewallu a vzdáleného přístupu. V další kapitole budou provedeny jednotlivé zátěžové testy pro různé služby a velikosti datových souborů. Výsledky realizovaných testů budou následně vyhodnoceny. Osmá kapitola se bude věnovat měření vlivu DDoS útoků na webové služby poskytované serverem. Konkrétně budou realizovány útoky SynFlood, UDPFlood, XMasTree, ResetFlood a ARPFlood. Z výsledků testování bude následně vybrán nejúčinnější DDoS útok, proti kterému budou v následující kapitole navrženy ochranné prostředky na úrovni „end-host“ ochrany, čili ochrany implementované na koncovém zařízení. Ochranné prostředky budou poté zavedeny na testovaný server a podrobeny vybranému DDoS útoku. Na základě získaných výsledků bude provedeno zhodnocení účinnosti jednotlivých ochran.

1 OPERAČNÍ SYSTÉM LINUX

Linux je častou volbou operačního systému, na kterém se má vybudovat síťový server. Část slávy Linuxu jako serverového systému pochází z jeho častého použití ve spojení s webovým serverem *Apache* [1]. Síla a spolehlivost Linuxu ovšem nabízí více než jen stabilní platformu pro jeden z nejpopulárnějších webových serverů. Poskytuje všechny důležité síťové služby, přičemž nízká cena, spolehlivost a efektivita pohánějí neustálý rozvoj Linuxu jako serverového systému. Toto vše značí, že tento systém je možné použít k uspokojení jakýchkoliv potřeb síťového serveru [2].

Z výše zmíněných důvodů je také na našem testovaném serveru použit operační systém Linux.

1.1 Historie Linuxu

Základ operačnímu systému Linux položil finský student Linus Torvalds. Na svém domácím počítači používal Unixový operační systém *Minix*[1]. Při jeho používání ovšem narazil na zásadní překážku – uzavřenost zdrojových kódů. V praxi to znamená, že nemohl měnit existující prostředí operačního systému, protože vlastnil pouze jeho binární verzi. Zdrojové kódy měli k dispozici jen samotní tvůrci. Řešení se naskytlo ve formě projektu GNU. Cílem tohoto projektu bylo vytvořit volně šířitelný klon Unixu, který by mohl být k dispozici včetně zdrojových kódů úplně každému [3]. Linus Torvalds tedy vytvořil prvotní jádro Linuxu založené na principech operačního systému Unix.

K jeho následnému vývoji a rozvoji využil Internet. Zdrojové kódy systému zpřístupnil právě prostřednictvím Internetu všem nadšencům, kteří se chtěli podílet na vývoji operačního systému. Do vývoje operačního systému Linux se pustili programátoři na celém světě a zároveň se v něm snažili implementovat jak vlastnosti operačního systému *Berkeley UNIX*[2], tak i vlastnosti operačního systému *System V UNIX*[3]. Přitom zde implementovali spoustu nových vlastností.

Operační systém Linux byl vytvořen a stále se vyvíjí za spoluúčasti mnoha lidí připojených k Internetu. Proto o něm lze říci, že je „produktem Internetu“ [4]. Během pouhých čtyř let se Linux změnil ze studentského pokusu přes výzvu na trhu serverů až na uznávaný systém, který zaujímá své právoplatné místo v akademických a firemních sítích [5].

[1] minimal-unix, napsal jej Andrew Tanenbaum k vyuce a knize o operačních systémech

[2] verze UNIX vyvinutá na kalifornské univerzitě v Berkeley

[3] vyvinut společnostmi AT&T a UNIX Systems Laboratories

1.2 Výběr distribuce

Linux sám o sobě je srdcem operačního systému: tzv. jádro, neboli kernel. Jádro je program, který funguje jako „manažer provozu“. Je odpovědné za spouštění a ukončování dalších programů, obsluhu žádostí o paměť, přístup k diskům a správu připojení k síti. Seznam činností jádra je opravdu obsáhlý, tudíž se nejedná o triviální program. Je tím, co se vkládá do všech distribucí Linuxu. Všechny distribuce používají totožné jádro, a proto je základní chování všech distribucí Linuxu stejné [6].

Linuxových distribucí je celá řada, přičemž jednotlivé distribuce se od sebe liší například nástroji, které jsou součástí každé z nich. Dále zejména způsobem použití[4] nebo skutečností, zda se jedná o komerční, či nekomerční distribuci.

1.2.1 Distribuce Debian

Jedná se o jednu z nejstarších doposud vyvíjených distribucí Linuxu. Je vyvíjená velkým množstvím dobrovolníků z celého světa a šířená jako nekomerční distribuce. Je velice rozšířená a známá především svou konzervativností [7]. Vychází z ní velká řada dalších úspěšných distribucí (např. *Ubuntu*).

Její použití je určeno především pro síťové servery a doporučuje se zkušenějším uživatelům. Samozřejmostí je možnost výběru mezi grafickým uživatelským rozhraním GUI (Graphic User Interface) nebo prostředím příkazové řádky LUI (Line User Interface). *Debian* obsahuje také vlastní balíčkovací systém APT (Advanced Packaging Tool) [8], který umožňuje jednoduchou správu balíčků z různých zdrojů.

Jednou z výhod distribuce *Debian* jsou její „větve“, do kterých se software člení podle úrovně otestování a míry funkčnosti [7]:

- **stable**: jedná se o stabilní, pečlivě otestovaný software, který je připravený pro nasazení i v kritických aplikacích. Bývá poněkud zastaralý, ale na druhou stranu pravidelně záplatovaný, má vyřešeny kritické chyby a bezpečnostní problémy. Využívá se převážně pro servery, kde preferujeme stabilitu a funkčnost.
- **testing**: testovací větev s novějším softwarem, avšak s možným výskytem chyb. Obsahuje balíčky, které zatím nebyly přidány do „stabilní“ verze.
- **unstable**: tato větev je označována jako „nestabilní“, vývojářská větev používaná převážně vývojáři a jinými nadšenci. Obsahuje nejnovější software, který nebývá důkladně odladěn.

Současná nejnovější „stabilní“ verze je *Debian 7.0 wheezy* (vydáno 4. května 2013). Aktualizace, technickou dokumentaci a podporu lze nalézt na oficiálních stránkách distribuce *Debian* `www.debian.org`.

[4] existují distribuce pro stolní počítače, servery, multimediální stanice apod.

2 SLUŽBY SERVERU

Služby, které nám poskytují webové a FTP servery, se staly běžnou součásti našich životů. Díky globálnímu rozšíření internetu se webové a datové služby stávají důležitou, ne-li hlavní formou získávání a zprostředkovávání informací či dat.

Webový server pracuje s protokolem HTTP (Hypertext Transfer Protocol), který je určen pro výměnu hypertextových dokumentů ve formátu HTML (HyperText Markup Language). Server je zodpovědný za vyřizování HTTP požadavků klienta a na jejich základě poskytuje příslušné odpovědi (nejčastěji např. HTML stránku).

FTP server pracuje s protokolem FTP (File Transfer Protocol), určeným pro přenos souborů. Používá se opět klasický model klient-server, kdy FTP server poskytuje data ostatním klientům v síti na základě jejich žádostí.

V rámci této práce nám HTTP a FTP služby poskytnou Apache a Vsftpd server.

2.1 Apache

Apache je nejpoužívanější softwarový HTTP server. V současnosti je díky němu v provozu 50 - 60 % všech aktivních webových stránek [9]. Jedná se o výkonný, flexibilní a kompatibilní webový server běžící na nejpoužívanějších operačních systémech. Tyto vlastnosti spolu s vysokou mírou konfigurovatelnosti, možnostmi rozšíření o vlastní moduly a otevřeným kódem patří mezi hlavní důvody jeho rozšířenosti a popularity. Díky tomu, že Apache podporuje velké množství funkcí formou rozšiřujících modulů, můžeme jej jednoduše doplnit o podporu PHP, MySQL, SSL apod. [10].

Apache je aktivně vyvíjen v rámci *Apache HTTP Server Project*, což je projekt řízený skupinou dobrovolníků z celého světa podílejících se na vývoji Apache a jeho dokumentaci.

2.2 Vsftpd

Jedná se o softwarový FTP server pro unixové systémy, včetně Linuxu. Zkratka programu Vsftpd znamená *Very Secure FTP Deamon*, neboli velice bezpečný FTP démon (program v neustále činnosti dlouhodobě běžící na pozadí systému). Program je vyvíjen jako svobodný software pod licencí GPL (GNU General Public License). Jedná se o bezpečný a velice rychlý FTP server, jehož další oceňovaná vlastnost je jeho stabilita. Díky těmto vlastnostem si našel oblibu u mnoha firem a organizací. Vsftpd je využíván například u: `ftp.redhat.com`, `ftp.suse.com`, `ftp.debian.org`, `ftp.gnu.org`, `ftp.gimp.org` a mnoho dalších [11].

3 TESTOVÁNÍ ZÁTĚŽE

Testování zátěže je poměrně široký pojem, se kterým se dnes často setkáváme u moderních technologií a jejich služeb. Obecně řečeno se jedná o řízený proces zatížení specifického zařízení, systému či služby a sledování následných reakcí na danou zátěž. Na základě vyhodnocení tohoto procesu jsou získány cenné informace o schopnostech, omezeních a funkčnosti daného objektu při zátěži, což je primární záměr realizace zátěžového testu. Díky tomu můžeme být schopni odpovědět si na důležité otázky typu: „Jaké chování můžeme od objektu očekávat při zátěži dosahující jeho mezí? Kde jsou jeho slabiny? Jakým způsobem předcházet vzniklým situacím?“ apod. V našem případě je objektem myšlen linuxový server s instalovanými webovými službami.

Existuje celá řada výkonnostních testů a možností, jakým způsobem objekt podrobit zátěži[1]. Konkrétně se zaměříme na možnosti testů simulující práci velkého počtu uživatelů či generující velký počet operací, transakcí a navázaných spojení. K samotné realizaci zmíněných možností využijeme zařízení Spirent Avalanche 3100.

3.1 Spirent Avalanche 3100

Jedná se o zařízení poskytující bezpečnostní, kapacitní a výkonové testování pro síťovou infrastrukturu, webové služby a služby zajišťující kvality služeb (QoS a QoE). Pracuje na vrstvách 4-7 referenčního modelu ISO/OSI. Dokáže simulovat jak provoz ze strany klientů, tak i ze strany serverů a to dokonce souběžně na různých portech. Umožňuje vytvořit až 30 miliónu spojení, specifikovat druhy spojení, identifikovat zpoždění vznikající v síti a ztrátovost paketů, testovat zranitelnost pomocí DDoS útoků a jiné. Součástí jsou samozřejmě i informace o průběhu simulací a jejich výsledky. Podporuje celou škálu protokolů na transportní[2] i aplikační[3] vrstvě, dále protokoly a kodeky pro podporu hlasu a videa nebo rozšířené protokoly komerčních subjektů[4]. Více informací, kompletní výčet funkcí zařízení a popis hardwaru lze nalézt v technické dokumentaci na stránkách výrobce `www.spirent.com`. Na popis nejdůležitějších funkcí a nastavení potřebných pro úspěšnou práci s tímto zařízením za účelem testování zátěže se podíváme podrobněji.

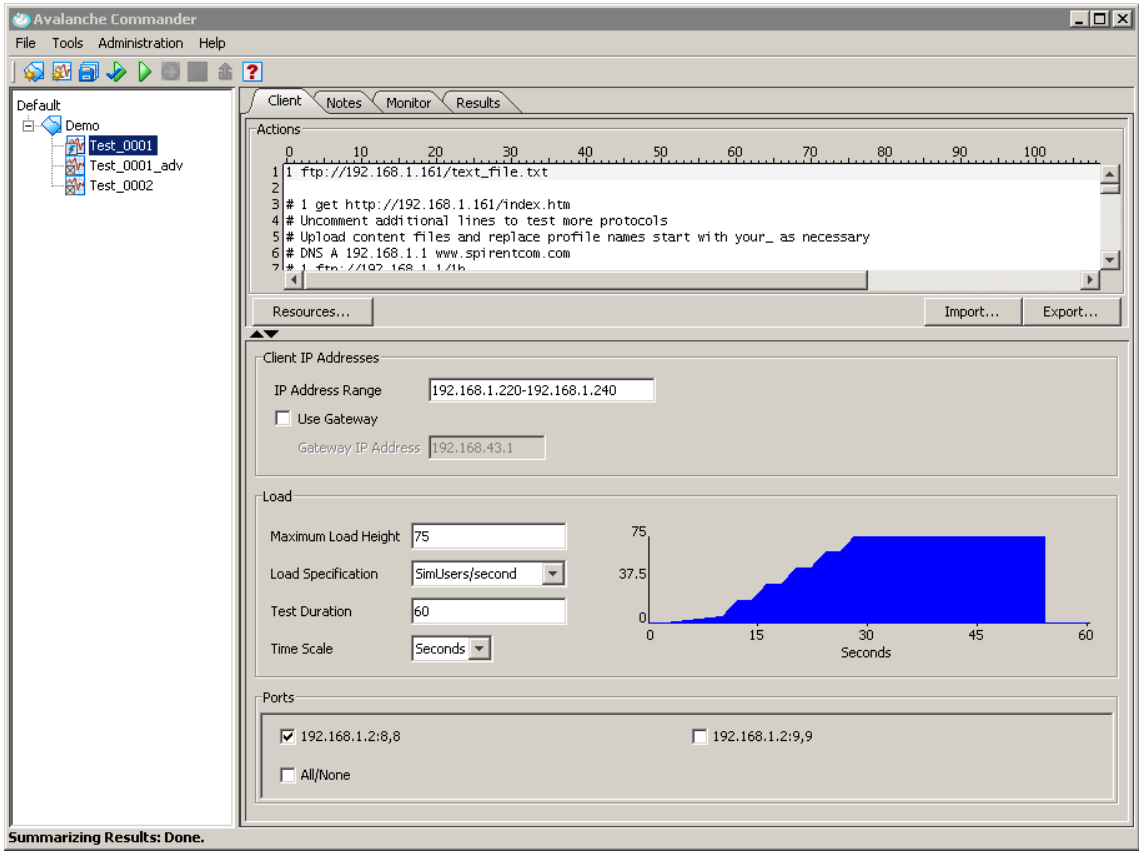
Pro samotnou práci se zařízením Avalanche využijeme dvě aplikace. *TestCenter Layer 4-7 Application* a *TestCenter Results Analyzer*. Teoretické znalosti získané o těchto aplikacích a jejich funkcích využijeme v kapitole zabývající se praktickou

[1] např. testování hraniční zátěže, odolnosti, nárazových špiček, citlivosti sítě, objemu dat...
[2] TCP, UDP, SSLv2, SSLv3...
[3] HTTP, HTTPS, FTP, DNS, TELNET, SMTP, IMAP4, POP3, ICMP...
[4] SKYPE, Oracle, MSN, Yahoo...

realizací zátěžových testů (kap. 7). Definice jednotlivých pojmů vychází z nápovědy těchto programů.

3.1.1 TestCenter Layer 4-7 Application

Jedná se o aplikaci sloužící k nastavení veškerých specifikací testu a k monitorování jeho průběhu. Před samotným testováním je potřeba nejdříve vytvořit nový profil. Na výběr máme ze dvou úrovní profilu. „Quick test“ a „Advanced test“. Při realizaci zátěžových testů (HTTP, FTP testy) budeme pracovat s prvním z nich. Vzhled aplikace v tomto profilu je vidět na obrázku 3.1. V levé části obrazovky aplikace



Obr. 3.1: Vzhled programu *TestCenter Layer 4-7 Application*.

můžeme přepínat mezi jednotlivými profily, které jsme vytvořili. V pravé části jsou záložky *Client*, *Notes*, *Monitor* a *Results*.

Nejpodstatnější záložka je *Client*, sloužící k nastavení specifik testu a je rozdělena do několika bloků. V bloku *Actions* je posloupnost příkazů, které jsou vykonané každým generovaným klientem. Můžeme zde využít celou řadu dostupných protokolů. V bloku *Client IP Addresses* nastavujeme rozsah IP adres, jenž jsou v průběhu zátěže přidělované nově generovaným klientům či požadavkům. Blok *Load* slouží

k nastavení specifikace zátěže. V poli *Load Specification* máme na výběr z těchto možností:

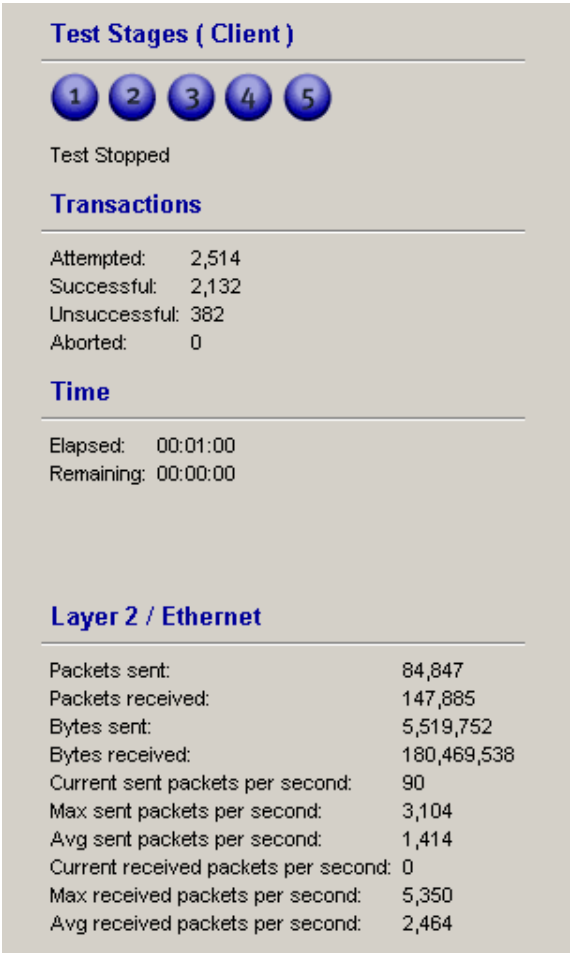
- **Bandwidth** – určuje množství dat, která mohou být přenesena přes rozhraní za stanovenou dobu. Obvykle se udává v kilobitech za sekundu.
- **BodyBytes** – generuje HTTP žádosti, které získávají od simulovaného serveru těla HTTP odpovědí dané velikosti. Tato možnost se využívá pouze pro simulované servery.
- **Connections** – označuje TCP spojení. Definuje počet současných síťových připojení inicializovaných z Avalanche. Je generována dostatečná zátěž k dosažení a udržení požadovaného počtu otevřených TCP spojení. Tuto zátěžovou specifikaci lze použít pro jakýkoliv protokol postavený na bázi TCP včetně HTTP, FTP a SMTP. Je dobré si taky uvědomit, že stejné množství otevřených TCP spojení může znamenat pro různé systémy různou velikost generované zátěže. Přes otevřené spojení totiž nemusí nutně probíhat neustála aktivita.
- **Connections/second (hours)** – je u výrobců síťového vybavení často preferovaná specifikace zátěže pro testování síťových zařízení. Jedno TCP spojení může obsahovat až stovky transakcí v závislosti na konfiguraci profilu klienta a odpovědí serveru. Avalanche dynamicky nastavuje rychlost příchozích uživatelů tak, aby to odpovídalo zadanému počtu spojení za sekundu (hodinu).
- **SimUsers** – jedná se o zkratku, označující simulované, virtuální uživatelé, kteří vykonají požadavek uvedený v *Action list*. Tato zátěžová specifikace opět generuje dostatečnou zátěž k dosažení a udržení požadovaného počtu souběžně simulovaných uživatelů. Používá se, když chceme generovat zátěž i v případě, že zařízení během testu selže. V tomto nastavení je důležité podotknout, že množství generovaného provozu je závislé na výkonu testovaného zařízení. To znamená, že pokud systém začne zpomalovat kvůli přetížení (každému uživateli začne trvat zpracování požadavků delší dobu), tak Avalanche sníží počet nově generovaných užvatelů.
- **SimUsers/second (hours)** – udržuje požadovaný počet souběžně simulovaných uživatelů za sekundu, čímž poskytuje realističtější simulaci generování zátěže na systém. V případě selhání systému totiž nesnižuje počet nově generovaných uživatelů, ale pokračuje v generování zátěže nezávisle na jeho chování.
- **Transactions** – definuje počet souběžně generovaných transakcí. Vytváří a udržuje dostatečné množství zátěže k dosažení zadaného počtu aktivních HTTP transakcí. Transakce v tomto smyslu znamená žádost a přenesení jednoho objektu, webové stránky. Jedna se tedy o specifikaci určenou pouze pro protokol HTTP nebo HTTPS.
- **Transactions/second (hours)** – určuje počet generovaných HTTP transakcí za sekundu, či hodinu. Opět v praxi více využívané.

Dále nastavujeme maximální váhu zátěže (*Maximum Load Height*), neboli množství zvolené specifikace. V neposlední řadě je zde také nastavení celkové doby běhu testu (*Test Duration*) a časového měřítka (*Time Scale*). Vidíme zde také graf zobrazující nárust zátěže v závislosti na čase. Graf se skládá z pěti časových úseků:

1. DELAY – zpoždění určené pro přípravu síťové karty (5 % doby testu).
2. RAMP UP – navýšení zatížení na počáteční úroveň (10 % doby testu).
3. STAIRS – postupný nárůst zátěže až na maximální úroveň (25 % doby testu).
4. STEADY – doba, během které je generování zátěže udržováno na požadované maximální úrovni (40 % doby testu).
5. RAMP DOWN – doba pro dokončení probíhajících operací (20 % doby testu).

Možnost nastavení trvání jednotlivých částí nabízí profil v pokročilé úrovni. V posledním bloku *Ports* vybíráme porty, ze kterých bude zátěž generována.

Další významnou záložkou je *Monitor*. Informuje o průběhu testu, v jaké fázi se nachází, počet aktuálně provedených pokusů a jejich úspěšnost, celkový a zbývající čas testu a průběžné statistiky počtu přenesených dat (obr. 3.2).



Obr. 3.2: Vzhled záložky *Monitor*.

Poslední dvě záložky *Notes* a *Results* umožňují uživateli zobrazení historie provedených testů a jejich popis.

Pro pozdější měření vlivu DDoS útoků na úspěšnost serveru v části 8.2 již budeme potřebovat parametry profilu „Advanced test“, který nám oproti profilu „Quick test“ nabízí podrobnější nastavení jednotlivých parametrů testů a především možnost realizace DDoS útoků. Základní nastavení testu je obdobné jako u výše uvedeného profilu, proto zde nebude podrobněji popisováno. Klíčové je ovšem nastavení parametrů DDoS útoků, které najdeme v záložce *Client -> Ports -> DDOS*. Zde se nám nabízí záložka *Attacks* pro výběr požadovaného útoku (podrobněji o poskytovaných útocích v části 4.1). Další záložkou je *Attack Variables*, která slouží pro nastavení parametrů vybraného útoku či útoků. Důležitými parametry jsou:

- *RepeatCount* – počet opakování útočné sekvence.
- *PacketToGenerate* – počet paketů vygenerovaných v každé útočné sekvenci.
- *PacketRate* – určuje počet generovaných DDoS paketů za sekundu.

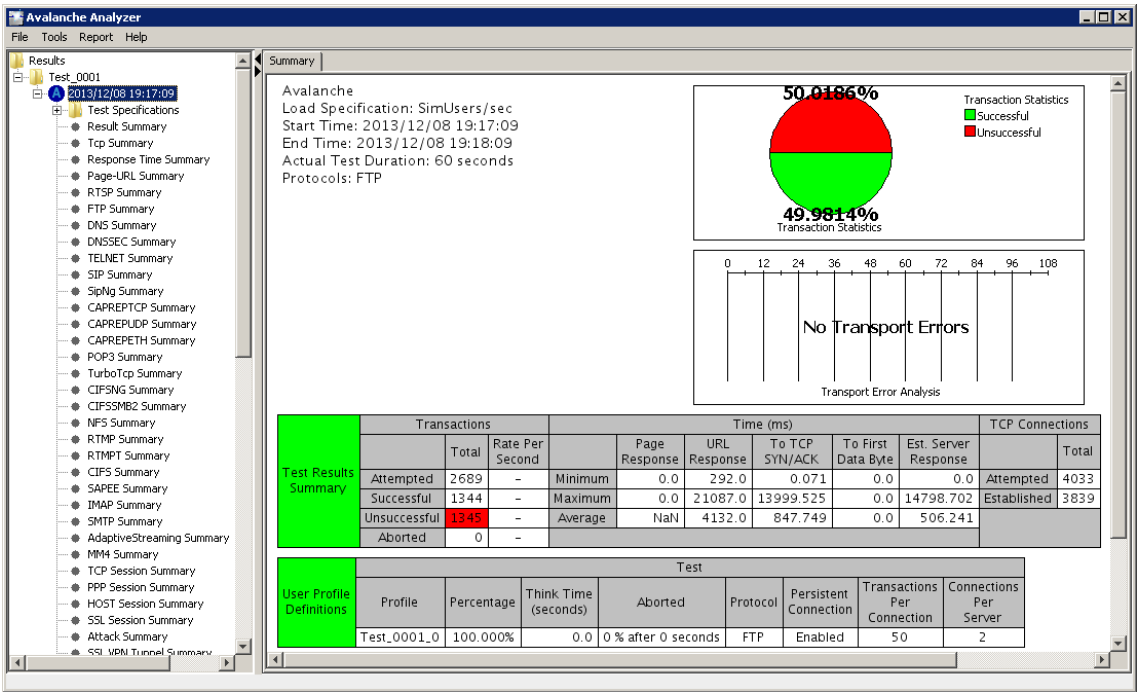
Poslední záložka nese název *Global Variables* a slouží pro globální nastavení všech útoků. Hlavními parametry jsou:

- *StartingSourceMACAddress* – zdrojová MAC adresa prvního DDoS paketu.
- *MACAddressIncrement* – určuje, zda se číslo zdrojové MAC adresy bude zvyšovat nebo snižovat a o jakou hodnotu.
- *StartingDestMACAddress* – cílová MAC adresa prvního DDoS paketu.
- *StartingSourceIPAddress* – zdrojová IP adresa prvního DDoS paketu.
- *StartingDestIPAddress* – cílová IP adresa prvního DDoS paketu.
- *GlobalStartDelay* – doba v milisekundách, za kterou se začne generovat DDoS útok po startu testu.

Výsledky testů (obou zmíněných profilů) lze zobrazit pomocí aplikace *TestCenter Results Analyzer*.

3.1.2 TestCenter Results Analyzer

Tato aplikace interpretuje výsledky z testování v přehledné grafické podobě (obr. 3.3). Hlavním globálním výsledkem je procento úspěšností provedených požadavků na testované zařízení. V levé části okna je pak možnost rozkliknout seznam dílčích výsledků testů, z níchž aktivní (tzn. obsahují nenulové výsledky) jsou ty, které odpovídají použitým protokolům a specifikaci daného zátěžového testu. Výsledky je možné dále exportovat do formátu pdf a html.



Obr. 3.3: Vzhled programu *TestCenter Results Analyzer*.

4 DDOS ÚTOKY

DoS útok, neboli spojení Denial of Service, v překladu znamená odmítnutí, či odepření služby. Často se také hovoří o tzv. DDoS útocích (Distributed Denial of Service), což jsou DoS útoky vedené z více zdrojů. Důsledkem těchto útoků je narušení nebo kompletní znepřístupnění služby pro její běžné uživatele. Nedostupnost služeb a systémů způsobuje jejich provozovateli značné finanční ztráty, poškození renomé firmy, či nutnost analyzovat a napravit vzniklý problém. Uvážime-li, že většina dnešních firem a organizací je na síťových službách značně závislá, máme co dočinění s velkým problémem.

Populárnost těchto útoků mezi útočníky značně roste. Pro jejich realizaci nejsou v dnešní době potřebné žádné hluboké znalosti, jelikož vhodné nástroje k jejich provedení jsou běžné dostupné. Fakt, že narušit činnost sítě nebo systému je často mnohem jednodušší než do ní získat přístup, je jen dalším důvodem jejich rozšíření [12].

Motivů k provedení útoku je hned celá řada. Může jít například o útoky na konkurenční společnosti s cílem poškodit její renomé, či jiným způsobem oslabit její pozici na trhu. V médiích se také často setkáváme s útoky, které jsou vedené jako určitá forma demonstrace. Cílem jsou v těchto případech nejčastěji vládní sítě a systémy, či skupiny a společnosti ohrožující svobodu slova. V neposlední řadě jsou DDoS útoky využívány jako teroristická hrozba.

V dnešní době existuje obrovské množství různých typů DDoS útoků. Velice často je využíváno nedostatků v architektuře a činnosti TCP/IP protokolů, které byly ve své době navrženy pro použití v otevřeném a důvěryhodném prostředí [12]. Útoky jsou dále zaměřované na obsazení přenosové kapacity linky, vyčerpání systémových prostředků, chyby v programech, na systémy směrování paketů, či DNS. Útoky lze také specifikovat a rozdělit podle TCP/IP vrstev, skrze které působí. V našem případě se zaměříme na stručnou analýzu těch DDoS útoků, které poskytuje zařízení Avalanche 3100.

4.1 Analýza dostupných útoku zařízení Avalanche

Zařízení Avalanche od společnosti Spirent Communications nabízí kromě zátěžového testování (kap. 3) také možnost generování DDoS útoků. Charakteristika jednotlivých typů útoků vychází ze zdrojů [13], [12] a z nápovědy programu *TestCenter Layer 4-7 Application*.

4.1.1 ARPFlood Attack

Jedná se o útok na počítačové sítě a síťová zařízení, pokoušející se zneužít jejich limitací v oblasti řízení ARP (Address Resolution Protocol) zpráv a jejich ukládání do mezipaměti. Zařízení Avalanche tímto útokem generuje ARP zprávy na cílené zařízení z rozsahu virtuálních zdrojových adres. Umožňuje také nastavit typ generovaných ARP zpráv (ARP žádost nebo ARP odpověď).

4.1.2 EvasiveUDP Attack

Útok generuje velký datový tok UDP datagramů s variabilní velikostí a náhodnou zdrojovou IP adresou. Pokud jsou datagramy zasílány ve velkém množství, mohou způsobit narušení systému oběti.

4.1.3 Land Attack

Jádro útoku spočívá v podvržení TCP SYN paketu tak, aby zdrojová IP adresa a zdrojový port paketu byl shodný s IP adresou a číslem portu zařízení oběti. Takto podvržené pakety jsou poté zaslány na zařízení oběti, které se začne pokoušet navazovat spojení samo se sebou, což může způsobit jeho zhroucení.

4.1.4 PingOfDeath Attack

Jedná se o zaslání ICMP (Internet Control Message Protocol) paketu s délkou větší než 65536 bajtů, což je maximum povolené specifikací TCP/IP. Pro přenos sítí je paket fragmentován a zaslán na příslušný cíl. Cílový systém se poté z přijatých fragmentů pokouší sestavit paket do původní, nepovolené velikosti, což může vést k přeplnění bufferu a zamrznutí, či zhroucení systému.

4.1.5 PingSweep Attack

PingSweep útok generuje ICMP ping žádosti na rozsah cílových adres sítě. Užívá se k identifikaci dostupných síťových zařízení v síti. Útočník si tak může vytvořit mapu aktivních síťových zařízení, na která lze zaútočit.

4.1.6 RandomUnreachableHost Attack

Útok spočívá v zasíláni ICMP chybových zpráv o nedostupnosti cíle na různá zařízení sítě. Záměrem útoku je přimět systém oběti, aby na základě chybových zpráv zahazoval navázána spojení.

4.1.7 ResetFlood Attack

Na zařízeni oběti je generována posloupnost TCP paketů, které mají za úkol resetovat, resp. ukončit, reálná aktivní TCP spojení.

4.1.8 Smurf Attack

Tento útok generuje ICMP paket s ping žádosti (echo request) na vysílací adresu nějaké sítě (např. broadcast adresu) s podvrženou adresou odesílatele, která odkazuje na cíl útoku. Všechny počítače v dané síti na ping odpoví a zahltí tak příslušný cíl. Síť tady funguje jako zesilovač původního požadavku.

4.1.9 SynFlood Attack

Tento útok využívá základního principu protokolu TCP při navazování spojení, tzv. „three-way handshake“. Při útoku je na cílové zařízení (např. server) vyslán SYN paket s podvrženou IP adresou. Server pro potencionální spojení vyhradí systémové prostředky a odpoví paketem SYN/ACK odesílateli na podvrženou IP adresu. Z té ovšem odpověď nepřichází a tak server posílá zprávu znovu. Po vypršení určité doby bez odezvy druhé strany server ruší alokaci systémových prostředků spolu se záznamem o původní inicializaci spojení.

V případě, že je na server vysláno velké množství těchto paketů, dojde k navýšení počtu „napůl“ otevřených, nedokončených spojení, čímž se vyčerpají systémové prostředky serveru a ten přestane obsluhovat nové legitimní žádosti o spojení.

4.1.10 TCPPortScan Attack

TCPPortScan útok generuje posloupnost TCP SYN paketů s různými cílovými porty protokolu TCP na zařízeni oběti. Záměrem je zjistit, na kterých portech jsou dostupné TCP služby. Skenování portů se také využívá za účelem vyčerpání systémových prostředků zařízení, což vede k znemožnění navazovat nová TCP spojení.

4.1.11 Teardrop Attack

Podobně jako u útoku Ping of Death se Teardrop Attack snaží docílit zhroucení cílového systému zasláním několika fragmentů, které se nedají korektně sestavit. Prakticky se jedná o fragmentovaný IP paket, jehož jednotlivé fragmenty se překrývají, čímž se vzájemně přepisují.

4.1.12 UDPFlood Attack

Útok generuje velké množství UDP datagramů, což spotřebovává šířku pásma sítě a znemožňuje síťovým zařízením navázat nová UDP spojení. Trik také spočívá v tom, že UDP datagramy jsou zasílány na náhodné transportní porty cílového zařízení. Zařízení se snaží zjistit, jaká aplikace na těchto portech naslouchá. Když zjistí, že žádná aplikace na daných portech nenaslouchá, zasílá ICMP odpověď o nedosažitelnosti cíle. Tuto režií musí zařízení provést pro každý zaslaný UDP datagram. Napadenému zařízení tak dojdou prostředky pro navázáni nových UDP spojení.

4.1.13 UDPPortScan Attack

Útok UDPPortScan generuje posloupnost UDP datových paketů s různými cílovými porty protokolu UDP na dané zařízení. Většinou se používá pro identifikaci UDP portů, na kterých jsou dostupné UDP služby. Skenování portů lze využít i podobným způsobem jako UDPFlood útok k vyčerpání systémových prostředků a následnému znemožnění navazovat nová UDP spojení.

4.1.14 UnreachableHost Attack

Tento útok zasílá na zařízení oběti ICMP chybové zprávy o nedostupnosti cíle. Záměrem útoku je přimět systém oběti, aby na základě chybových zpráv zahazoval navázána spojení s daným cílem.

4.1.15 XmasTree Attack

Útok XmasTree generuje posloupnost TCP paketů s některými nastavenými příznaky (URG, PSH a FIN zároveň) používaných v TCP hlavičce. Pakety jsou zaměřený na cíl útoku a generovány z velkého počtu IP adres. Je tak využívaná chyba, která existuje v některých síťových zařízeních. Následkem je selhání systému po přijetí XmasTree paketu.

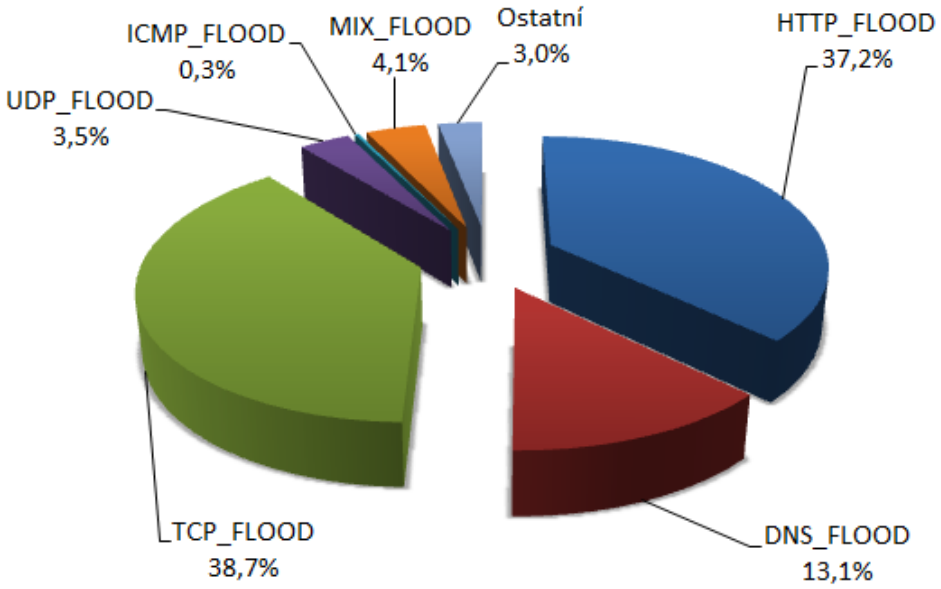
4.2 Analýza současného stavu DDoS útoků

Orientace v oblasti současného stavu a vývoje DDoS útoků je z hlediska preventivní ochrany značně klíčová. Velice užitečné jsou v tomto směru bezpečnostní závěrečné zprávy společností, které analyzují datový provoz internetové sítě a vyvíjejí bezpečnostní ochrany proti DDoS útokům. Tyto bezpečnostní zprávy nám mohou poskytnout cenné informace o trendech, metodách a taktikách DDoS útoků, jenž v současné době představují největší hrozbu. Následující text vychází z aktuálních

bezpečnostních zpráv za rok 2013 třech globálních společností[1] zabývajících se touto problematikou [14], [15], [16].

4.2.1 Nejčastější typy útoků

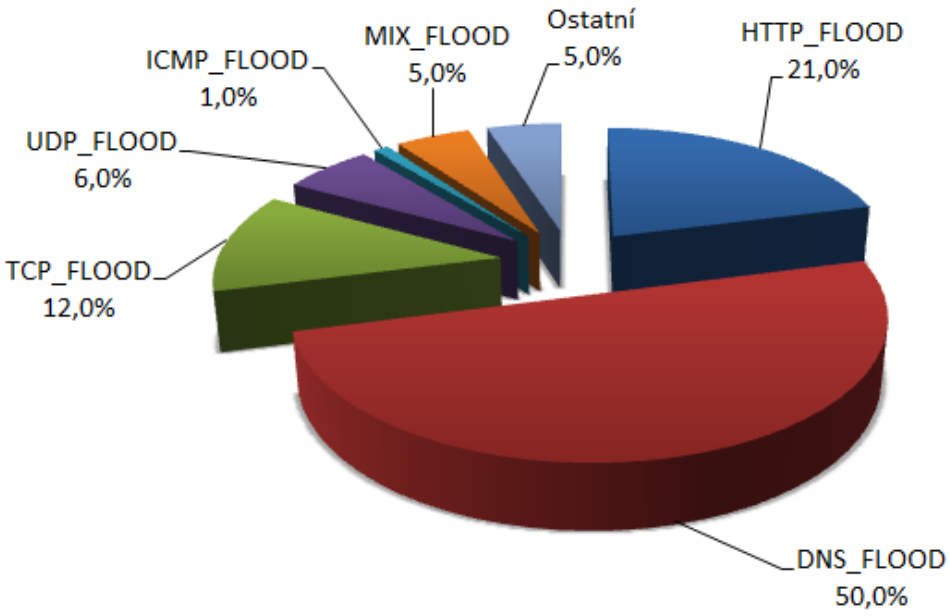
Od roku 2012 až do poloviny roku 2013 byly nejčastějšími útoky TCP flood (označován také jako SynFlood) a HTTP flood. Popularita těchto útoků se v druhé polovině minulého roku náhle snížila a převládajícím útokem se stal DNS flood, který tvoří více než 50 % z celkového počtu útoků uskutečněných v druhé polovině roku 2013. Tato změna se přisuzuje zvýšeným schopnostem ochran proti útokům TCP flood a HTTP flood a jejich stále častější implementaci v podnicích, hostingových společnostech a datových centrech. Důsledkem je tedy to, že se počítačoví zločinci zaměřují na DNS infrastrukturu, která zůstává jedním z nejslabších článků sítě a je tedy náchylná k DDoS útokům. Procentuální zastoupení nejčastěji používaných DDoS útoků je na obrázcích 4.1 a 4.2. První z nich je výsledkem analýzy za první polovinu roku 2013, druhý z nich odpovídá stavu v druhé polovině roku 2013.



Obr. 4.1: Nejčastější typy DDoS útoků (1. pol. 2013) [15].

V minulém roce byl také zaznamenám nárůst hybridních útoků, které se skládají z útoků založených na různých slabinách různých protokolů. Nejpopulárnějším se stal hybridní útok založený na protokolu TCP. Druhý nejčastější hybridní útok je založen na protokolech ICMP, TCP a UDP. Třetí místo pak obsadil útok využívající

[1] Radware, Prolexic Technologies a NSFOCUS Information Technology



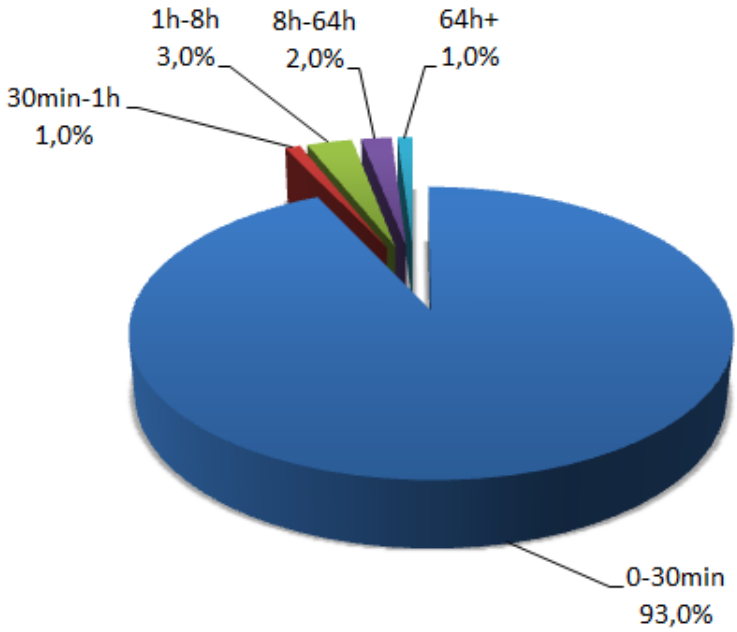
Obr. 4.2: Nejčastější typy DDoS útoků (2. pol. 2013) [15].

současně slabiny protokolů ICMP, TCP, UDP a DNS. Zároveň se ukazuje skutečnost, že DDoS útoky jsou používány nejen jako prostředek k narušení určité služby, ale stále častěji také jako určitá návnada, která má za úkol odvést pozornost od jiného útoku či krádeže dat.

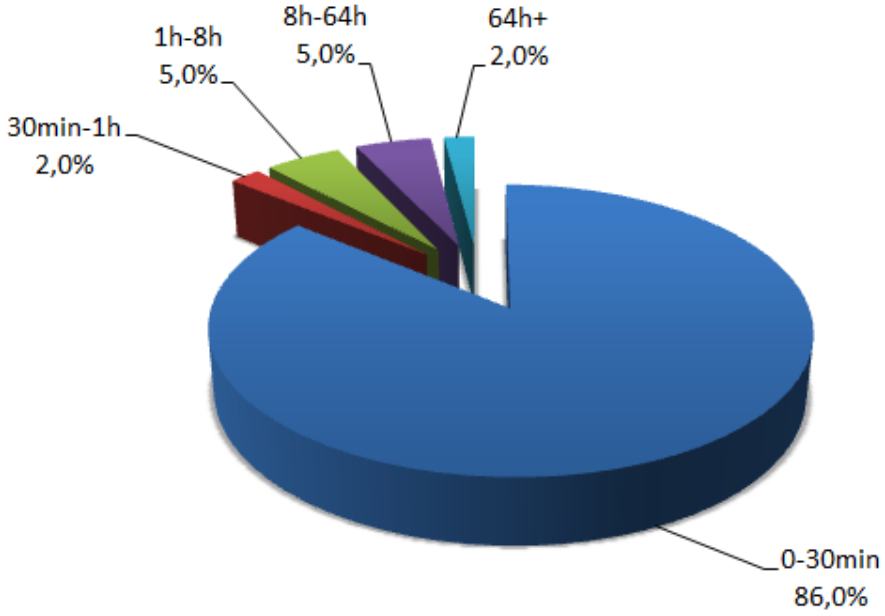
4.2.2 Doba trvání útoků a jejich velikost

Doba trvání většiny DDoS útoků v roce 2013 byla 30 minut nebo méně. Podobný trend byl zaznamenám také v roce 2012. Útoky trvající méně než 30 minut tvořily 93,2 % všech útoků zaznamenaných v první polovině roku 2013, v druhé polovině roku to pak bylo 86,2 %. Kromě toho narostl také počet útoků, které byly kratší než 15 minut, a to o téměř 50 % při srovnání první a druhé poloviny roku. Tento jev je přisuzován zkracování doby reakce na probíhající útok díky systémům na odklonění datového provozu či ochran proti DDoS útokům. Nutno ovšem podotknout, že nejsilnější a nejzávažnější zaznamenané útoky spadají do kategorie s dobou trvání 8 hodin a více. V této kategorií byl během roku 2013 zaznamenán nárůst o 4 %. Výsledky trvání DDoS útoků vidíme na obrázcích 4.3 (pro první polovinu roku 2013) a 4.4 (pro druhou polovinu roku 2013).

Co se týká velikosti datového toku DDoS útoků, nejčastější jsou útoky s velikosti do 50 Mb/s. Jedná se o 80 % všech útoků. Většinou se jedná o útoky na aplikační vrstvu nebo o hybridní útoky, které mají potenciál způsobit podstatné narušení



Obr. 4.3: Doba trvání DDoS útoků (1. pol. 2013) [15].

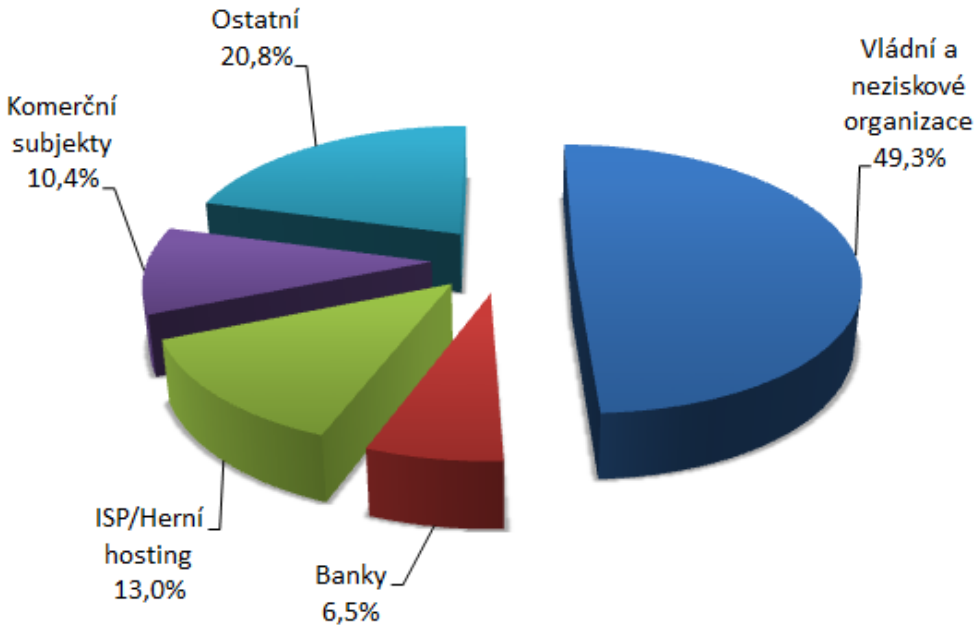


Obr. 4.4: Doba trvání DDoS útoků (2. pol. 2013) [15].

u systémů, které na podobné typy útoků nejsou připraveny.

4.2.3 Nejčastější cíle útoků

Nejvíce napadané cíle v průběhu roku 2013 lze prakticky rozdělit do dvou hlavních skupin: vládní sítě a online služby. V první z nich je motivace útoků založena na politických či ideologických motivech, u druhé skupiny je hlavním motivem finanční zisk, konkurenční rivalita nebo dokonce vydírání. Sledováním DDoS útoků v celosvětovém měřítku se zjistilo, že v první polovině roku 2013 byly nejčastějším cílem banky a jejich služby (43,3 % všech útoků), zatímco v druhé polovině roku obsadily první místo vládní a neziskové organizace (49,3 % všech útoků). Pro srovnání, útoky cílené na banky klesly na 6,5 %. Jedno z možných vysvětlení tohoto jevu je, že bankovní sektor od druhé poloviny roku podstatně posílil svou DDoS ochranu po prvních třech proběhlých etapách masivních DDoS útoků *Operace Ababil* (o tomto útoků, který proběhl v USA a je považován za jeden z nejzávažnějších světových útoků vůbec, podrobněji pojednává bezpečnostní zpráva společnosti Radware [14]). Nejčastější cíle útoků pro druhou polovinu roku 2013 jsou přehledně zobrazeny na obrázku 4.5.



Obr. 4.5: Nejčastější cíle DDoS útoků (2. pol. 2013) [15].

Z hlediska útoků na zařízení se nejčastěji setkáváme s útoky na prvky páteřních sítí, firewaly a servery.

Zajímavým poznatkem je také skutečnost, že více jak 50 % obětí DDoS útoků jsou napadeny více než jednou. Většinou se jedná o dvě až deset napadení během kalendářního roku.

4.3 Obecný návrh ochranných prostředků

Samotným útokům zabránit nemůžeme, v dnešní době ale existuje řada opatření a nástrojů, jak útokům předcházet, případně jak škody co nejvíce minimalizovat.

Na počátku návrhu ochraných prostředků je nutné stanovit si jednotlivé cíle, to znamená analýzovat jaké riziko pro nás DDoS útok představuje a jaké finanční, či jiné škody mohou být jeho důsledkem [17]. Na základě toho je potřeba zvážit, jaké množství zdrojů (finance, čas, lidské zdroje) jsme ochotni do opatření investovat. Při samotném návrhu a implementaci ochranných prostředků pak můžeme postupovat směrem od jednotlivých systémů, přes ochranu vlastní sítě, až po spolupráci s poskytovatelem připojení a Národním centrem kybernetické bezpečnosti.

Proti jednotlivým typům útoků bývají na úrovni zařízení či systémů navrženy různá opatření, jak se jim do značné míry bránit. Příkladem může být nastavení SYN cookies na daném zařízení, či systému, jako opatření proti SYN Flood útoku (RFC 4987 [18]). Podobná nastavení můžeme aplikovat i pro další typy útoků. Tato jednotlivá opatření není dobré podceňovat, jelikož cílem útoku může být jakýkoliv prvek síťové infrastruktury. Samozřejmostí by měl být vždy aktualizovaný a záplatovaný systém splňující příslušná RFC doporučení.

V rámci ochrany vnitřní sítě můžeme sáhnout po specializovaných síťových prvcích, které mají za úkol analýzovat síťový provoz, a v případě útoku jej včas detekovat a zastavit, či omezit. Mezi tato zařízení patří především různé aplikační firewally, systémy pro prevenci průniku IPS (Intrusion Prevention Systems), či IDPS (Intrusion Detection and Prevention Systems) a jiné. Na vývoj a prodej těchto zařízení se dnes zaměřuje řada firem. Žádné poskytované řešení ale není stoprocentní a je potřeba zvážit, který produkt od kterého výrobce je svou specifikací pro danou síť nejvhodnější (pokud vůbec). Důležitým aspektem při boji proti DDoS útokům je jejich odlišení od legitimního provozu, což nemusí být vždy tak jednoduché. Mnohdy totiž může i validní uživatelský provoz vykazovat charakteristiky DDoS útoků. Příkladem může být časově omezená slevová akce oblíbeného mezinárodního webového portálu, který se rázem stane cílem obrovského množství uživatelů (v principu se jedná o HTTP flood útok).

Následnou záležitostí pro ochranu před DDoS útoky je vzajemná spolupráce s poskytovatelem připojení k internetu. Sjednání určitých krizových postupů a opatření napomáhá k zamezení velké části útoku. Na ochraně proti masivním útokům se bojuje i na národní a nadnárodní úrovni. Ukázkou může být například český národní internetový uzel NIX.CZ, který pracuje na ochraně před velkými DDoS útoky a plánuje výraznější zabezpečení tuzemské internetové sítě [19]. Dalším takovým příkladem je NCKB (Národní Centrum Kybernetické bezpečnosti), které před nedávnem[2]

[2]Listopad 2013

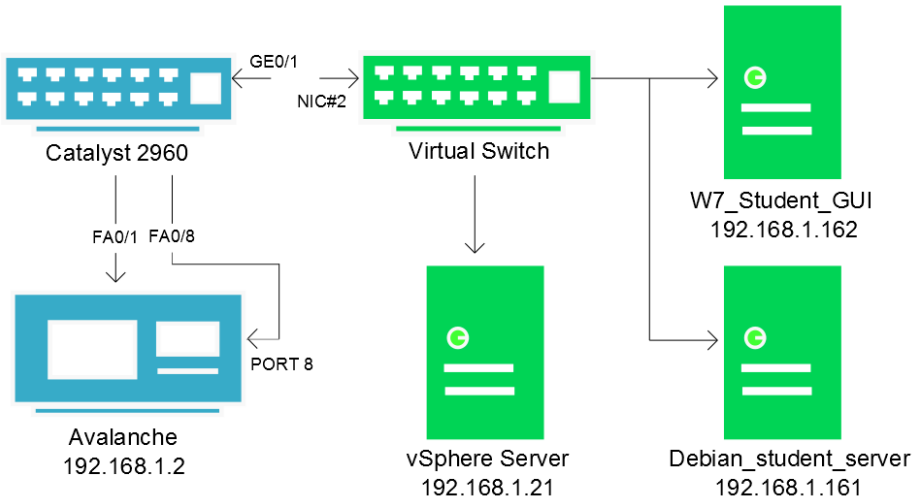
zveřejnilo dokument výstižně nazvaný „Doporučení pro případ napadení DDoS útokem – jak se zachovat a jak postupovat“, který se mimo jiné věnuje spolupráci mezi obětí a bezpečnostním pracovištěm CERT/CSIRT [20].

Každá síťová infrastruktura je svým způsobem jedinečná. Při konkrétním návrhu opatření proti DDoS útokům je tedy potřebná hluboká znalost problematiky a orientace v dané infrastruktuře. Vždy je zároveň potřebné zvážit, jaké množství zdrojů jsme ochotni do opatření investovat a zda nám to přinese kýžený výsledek. Ochraně proti DDoS útokům je ale v dnešní době potřeba vždy věnovat zvýšenou pozornost.

5 LABORATORNÍ SÍŤ

Pro účely zátěžového testování bez DDoS útoků (kap. 7) byla vytvořena experimentální síť využívající prostředků virtualizační platformy VMware vSphere. Jádrem platformy je výkonný hypervizor běžící na hostitelském hardwaru, umožňující provoz virtuálních počítačů. Těm pak přiděluje hardwarové prostředky jako procesor, paměť, disky apod. Další důležitou funkcí je tvorba virtuálních ethernetových přepínačů, které umožňují vzájemnou komunikaci mezi jednotlivými virtuálními počítači. Komunikace s vnějším světem je pak možná skrz přidělenou síťovou kartu hostitelského systému [21].

Na obrázku 5.1 vidíme reálné zapojení experimentální sítě. Zařízení vyznačená zelenou barvou jsou virtuální zařízení pracující na platformě VMware vSphere. Je zde testovaný server běžící na linuxové distribuci Debian (Debian_Student_Server) a pracovní stanice se systémem Windows 7 (W7_Student_GUI), na které jsou nainstalovány aplikace *TestCenter Layer 4-7 Application*, *TestCenter Results Analyzer* pro ovládání zařízení Avalanche a program *Putty* pro vzdálené připojení k linuxovému serveru. Dále modrou barvou jsou vyznačena zařízení fyzická. K přepínači



Obr. 5.1: Topologie laboratorní sítě pro zátěžové testování bez DDoS útoků.

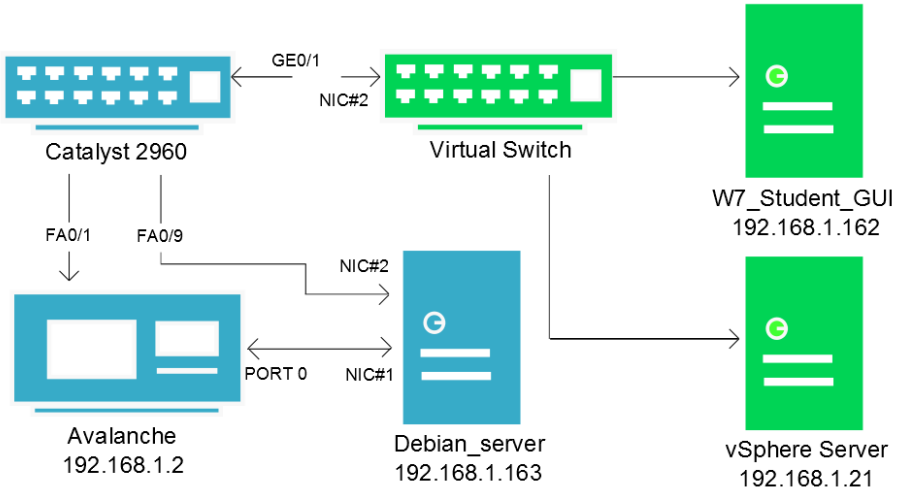
Catalyst 2960 je připojeno zařízení VMware vSphere (s virtuálními systémy) a zařízení Avalanche (dvě připojení). Jedno připojení k zařízení Avalanche slouží k jeho ovládání, druhé slouží ke generování zátěže – připojeno na port 8. Zařízení mají adresy přiděleny z privátní sítě 192.168.1.0/24.

Při realizaci zátěžových testů s DDoS útoky (kap. 8) bude topologie laboratorní sítě pozměněna. Testovaný server se již nebude nacházet ve virtualizační platformě

VMware vSphere, ale bude se jednat o samostatné fyzické zařízení. Jako testovaný server nám poslouží zařízení firmy Hewlett-Packard vybavené dvěmi síťovými kartami s teoretickou přenosovou rychlostí 1 Gbit/s. První z nich (NIC#1) slouží pro připojení k zařízení Avalanche na port 0, ze kterého se bude generovat požadovaná zátěž. Druhá síťová karta (NIC#2) je připojena k přepínači Catalyst 2960. Toto připojení bude sloužit ke vzdálenému připojení na testovaný server. V tabulce 5.1 jsou uvedeny základní hardwarové parametry testovaného zařízení. Pozměněná topologie laboratorní sítě pro zátěžové testování s DDoS útoky je na obrázku 5.2.

Tab. 5.1: Hardwarové parametry serveru.

Procesor	Intel(R) Xeon(R) CPU 3040 @ 1.86GH
Počet jader	2
Paměť RAM	4096 MB
Pevný disk	160GB
Síťové rozhraní NIC#1	HP PCIe Gigabit Server
Síťové rozhraní NIC#2	Intel PRO/1000 PT Desktop



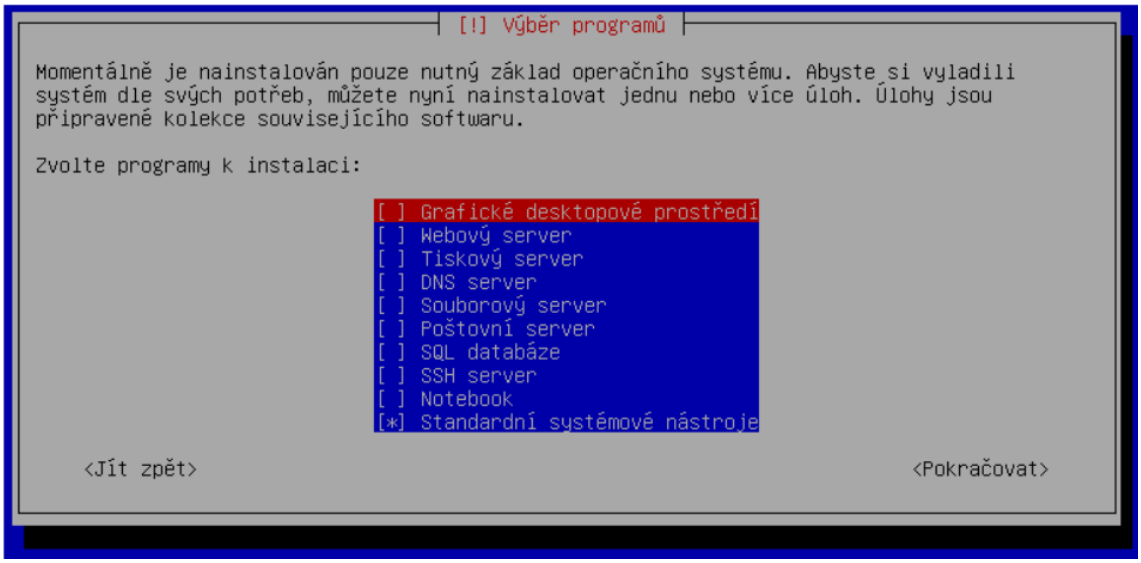
Obr. 5.2: Topologie laboratorní sítě pro zátěžové testování s DDoS útoky.

6 INSTALACE A KONFIGURACE PROSTŘEDÍ

6.1 Operační systém Linux

Jak již bylo zmíněno v kapitole 1, na testovaném serveru je použit operační systém Linux, konkrétně distribuce Debian. Vybrána byla nejnovější „stable“ verze systému[1] s označením *Debian 7.4.0 wheezy* (vydáno 8. února 2014) z oficiálních stránek distribuce Debian[2]. Cílem bylo nainstalovat pouze nejnutnější minimum pro funkčnost serveru. Z toho důvodů byla zvolena síťová instalace pomocí „minimal CD“ obrazu (určen pro 64-bitové systémy), který obsahuje pouze základ systému a volitelné části nebo aplikace umožňuje uživateli přidat dle jeho uvážení. Nutností je tedy připojení k internetu během instalace.

Samotná instalace probíhá intuitivním způsobem pomocí průvodce. V kroku instalace „Výběr programů“ (obr. 6.1) je ovšem potřeba dát si pozor na položku „Grafické desktopové prostředí“. Pokud položku neodznačíme, nainstaluje se nám



Obr. 6.1: Instalační okno pro výběr programů.

grafické rozhraní spolu s aplikacemi, které nechceme. Výsledkem instalace je tedy uživatelské rozhraní příkazové řádky.

[1] ke dni 17. 2. 2014

[2] `www.debian.org`

6.2 Webové a datové služby

Jako FTP server nám slouží aplikace Vsftpd (Very Secure FTP Daemon) a jako HTTP server aplikace Apache. Jejich domovské adresáže jsou v `/home/student/ftp` a `/home/student/www`. Veškeré níže zmíněné operace jsou prováděny s oprávněním uživatele root.

Instalace programu Vsftpd lze vyvolat jednoduchým příkazem `apt-get install vsftpd`. Po instalaci je program ve výchozím nastavení, čili umožňuje připojení anonymních uživatelů na FTP server. Aby mohli uživatele stahovat data, je potřeba nejdříve vytvořit kořenový adresář pro FTP server a do něj zkopírovat požadované soubory. Adresář vytvoříme známým příkazem `mkdir /home/student/ftp` a zkopírujeme do něj soubory. Nejjednodušším způsobem je soubory zkopírovat z připraveného datového nosiče. V případě, že chceme soubory kopírovat ze vzdáleného zařízení pomocí již vytvořeného FTP serveru, je potřeba pro něj povolit možnost zápisu a přihlašování registrovaných uživatelů. To provedeme odkomentováním záznamů `write_enable=Yes` a `local_enable=Yes` v konfiguračním souboru `/etc/vsftpd.conf`. Po zkopírování je potřeba záznamy opět zakomentovat, aby případní uživatelé nemohli měnit obsah serveru. Adresář se soubory poté přiřadíme FTP serveru jako kořenový. V konfiguračním souboru `/etc/vsftpd.conf` vytvoříme záznam `anon_root=/home/student/ftp` a uložíme. Nakonec program restartujeme příkazem `/etc/init.d/vsftpd restart`. Kořenový adresář musí mít nastavena práva pouze pro čtení, jinak nám program při restartu nahlásí chybu. Nyní je možné se k FTP serveru přihlásit direktivou *ftp://anonymous@ip_adresa_serveru* a stahovat z něj soubory.

Instalace a konfigurace programu Apache probíhá podobným způsobem. Nainstalujeme program příkazem `apt-get install apache2`, vytvoříme známým způsobem adresář `www` v adresáři `/home/student`, zkopírujeme do něj testovanou webovou stránku a přiřadíme jej jako kořenový adresář HTTP serveru. To provedeme v souboru `/etc/apache2/sites-available/default` v záznamu `DocumentRoot` změnou adresáře na `/home/student/www`. Pro funkčnost webové stránky je potřeba povolit právo čtení a spouštění u kořenovéo adresáře a jeho obsahu. To se provede příkazem `chmod 755 /home/student/www -R`. V posledním kroce HTTP server restartujeme `/etc/init.d/apache2 restart`.

6.3 Vzdálený přístup a firewall

K zabezpečenému vzdálenému přístupu na linuxový server nám slouží OpenSSH server. Jeho instalaci provedeme příkazem `apt-get install openssh-server`. Server poté standartně naslouchá na portu 22 transportní vrstvy. Na stanici s operačním

systémem Windows 7, ze kterého vzdálený přístup realizujeme, je jako klient použit program Putty.

Zabezpečení síťového provozu a pravidla pro komunikaci serveru v síti určuje firewall FireHOL. Instaluje se opět stejným způsobem jako předchozí balíčky příkazem `apt-get install firehol`. Na rozdíl od výše zmíněných programů, není po instalaci FireHOL aktivní. Pro jeho spuštění je potřeba editovat soubor `firehol` v aresáři `/etc/default/`, kde změníme záznam `START_FIREHOL=NO` na `START_FIREHOL=YES`. Samotná pravidla síťového provozu se zadávají v konfiguračním souboru programu `/etc/firehol/firehol.conf`. Bezpečnostní politika firewalu je restriktivní, což znamená, že co není přímo povoleno, je zakázáno. V našem případě chceme na všech síťových zařízeních povolit pro příchozí směr protokoly HTTP, FTP a SSH. Pro odchozí směr necháme povolenou veškerou komunikaci. V konfiguračním souboru tedy zapíšeme následující pravidla

```
interface any world
   server http accept
   server ftp accept
   server ssh accept
   client all accept
```

a restartujeme program příkazem `/etc/init.d/firehol restart`. Tímto jsme si připravili linuxový server určený pro zátěžové testování. Nutno ovšem zmínit, že v rámci zatěžového testování s DDoS útoky (kapitola 8), bude firewall FireHOL deaktivován.

7 ZÁTĚŽOVÉ TESTOVÁNÍ BEZ DDOS ÚTOKŮ

V rámci testování podrobíme zátěži nejdříve HTTP server a budeme sledovat jeho chování a úspěšnost provedených transakcí se zvyšující se zátěži. Poté podrobíme zátěži FTP server. Zjistíme kde jsou jeho limity a provedeme také srovnání pro různé velikosti stahovaných dat.

Dříve než začneme provádět testy, je nutné vytvořit v programu *TestCenter Layer 4-7 Application* nový profil. Ve vytvořeném profilu následně zrealizujeme výše zmíněné zátěžové testy. Popis prostředí aplikace a význam jednotlivých typů nastavení je popsán v části 3.1.1. Nyní se již zaměříme na konkrétní nastavení hodnot pro úspěšnou realizaci testu.

7.1 HTTP test úspěšnosti

Cílem testu je zjistit, při jaké zátěži bude server dosahovat úspěšnosti: 90, 80, 70, 60 a 50 procent. V záložce *Client* tedy nastavíme:

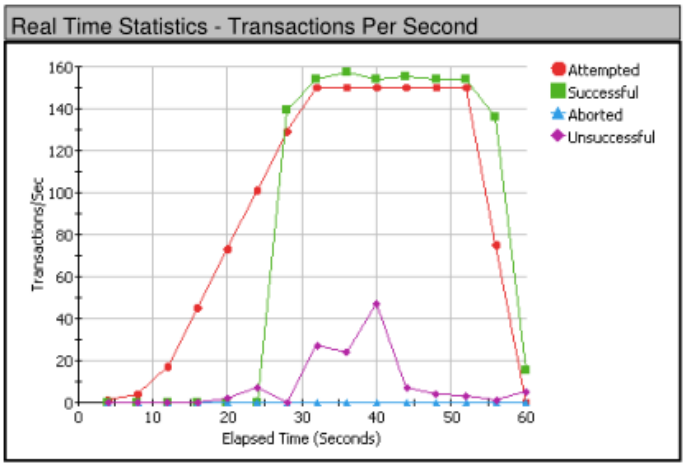
- *Actions*: `get http://192.168.1.161/index.htm`
- *IP Adress Range*: 192.168.1.220-192.168.1.240
- *Load Specification*: Transactions/second
- *Test Duration*: 60
- *Time Scale*: Seconds
- *Ports*: 192.168.1.2:8,8

a zátěž, položku *Maximum Load Height*, budeme v jednotlivých testech postupně navyšovat.

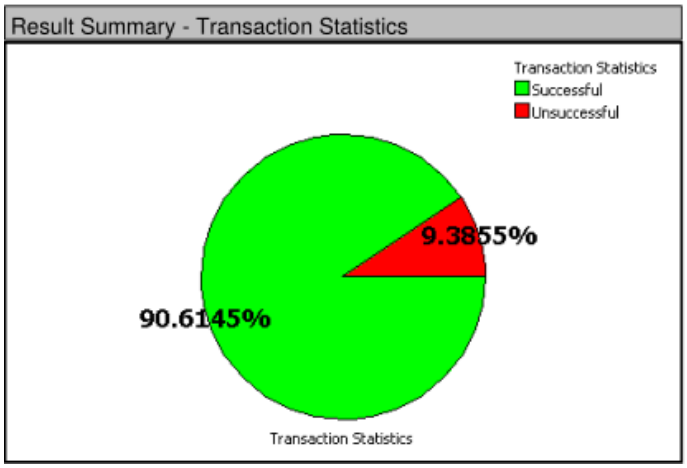
7.1.1 Výsledky pro úspěšnost 90 %

Při nastavení zátěže na **150 transakcí za sekundu** klesla úspěšnost provedených transakcí na 90,6 %. Celkem proběhlo 5 370 transakcí, z toho úspěšných bylo 4 866 a neúspěšných 504. Předčasně ukončena transakce nebyla žádná.

Ná obrázku 7.1 je graf znázorňující závislost počtu transakcí za sekundu na uplynulém čase. Vykresluje celkem 4 barevně odlišeny grafické závislosti (dle legendy: vyslané, úspěšné, předčasně ukončené a neúspěšné transakce). Sumarizované výsledky testu jsou na obrázcích 7.2 a 7.3. Užitečný je také výpis z logovacího souboru HTTP serveru umístěný v `/var/log/apache2/access.log`, kde můžeme sledovat z jakých IP adres jsou požadavky generovány a s jakou četností. Výpis z logovacího souboru je na obrázku 7.4.



Obr. 7.1: Graf závislosti počtu transakcí za sekundu na uplynulém čase.



Obr. 7.2: Sumarizované výsledky: procento úspěšnosti prvního testu.

7.1.2 Výsledky pro úspěšnost 80 %

Úspěšnosti 79,2 % bylo dosaženo při nastavení zátěže na **190 transakcí za sekundu**. Proběhlo 6 766 operací, z toho 5 357 úspěšných a 1 409 neúspěšných.

7.1.3 Výsledky pro úspěšnost 70 %

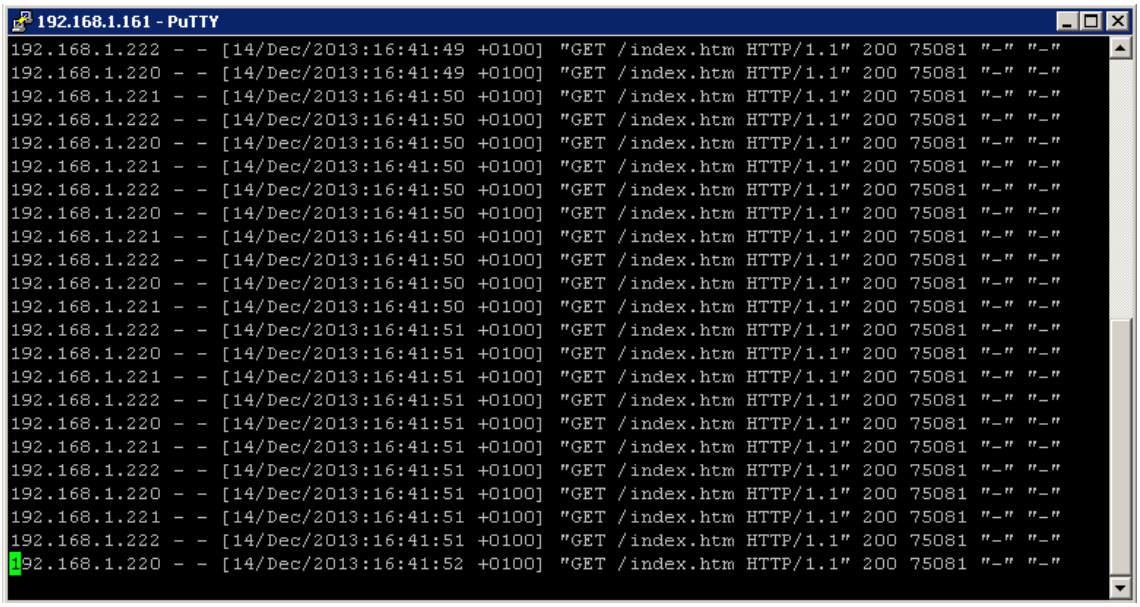
Při nastavení zátěže na **220 transakcí za sekundu** klesla úspěšnost provedených transakcí o dalších deset procent na hodnotu 69,1 %. Zvýšil se počet celkových transakcí na 7 873. Úspěšných bylo 5 437 a neúspěšných 2 436.

Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	5370	89	Minimum	6.0	6.0	1.012	0.516	0.0	Attempted	5370
	Successful	4866	81	Maximum	28924.0	28924.0	13930.278	14427.192	13158.442	Established	5085
	Unsuccessful	504	8	Average	1949.0	1949.0	1003.225	1073.311	780.567		
	Aborted	0	0								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Test_0001_0	100.000%	0.0	0 % after 0 seconds	HTTP 1.1	Enabled	50	2

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted
	Test_0001_0	1	81	5370	4866	504	0	90.61	9.38	0.0
	Totals	1		5370	4866	504	0	90.61	9.38	0.0

Obr. 7.3: Sumarizované výsledky pro úspěšnost 90,6 %.



Obr. 7.4: Výpis z logovacího souboru HTTP serveru.

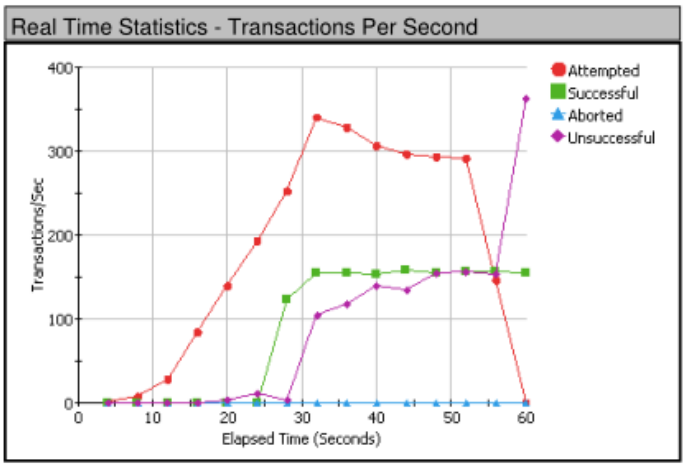
7.1.4 Výsledky pro úspěšnost 60 %

Další pokles na 57,9 % proběhl při zátěži **250 transakcí za sekundu**. Proběhlo 9 434 transakcí, z toho 5 462 úspěšných a 3 972 neúspěšných.

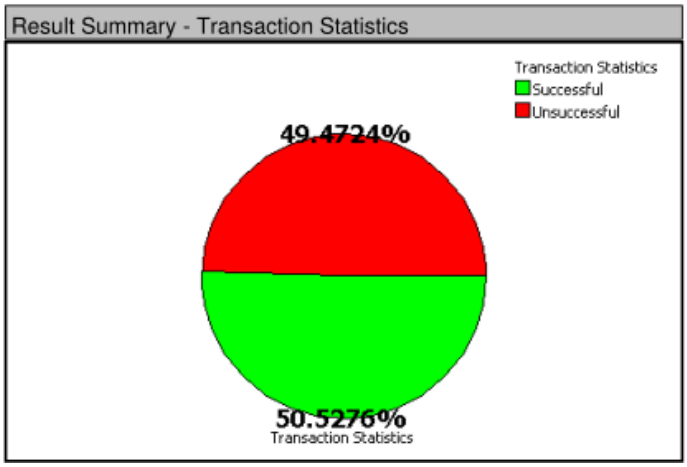
7.1.5 Výsledky pro úspěšnost 50 %

Při posledním testování byla zátěž nastavena na **290 transakcí za sekundu**. Dosáhlo se tak úspěšnosti 50,5 %. Během doby testu proběhlo 10 804 transakcí, přičemž 5 459 z nich bylo úspěšných a 5 345 neúspěšných.

Na obrázku 7.5 je opět známý graf s počtem jednotlivých transakcí za sekundu v závislosti na čase. Sumarizované výsledky jsou na obrázcích 7.6 a 7.7. Během zátěžového testu byl také pořízen záznam vytížení procesoru serveru ve správci procesů (obr. 7.8). Za užitečné můžeme označit také výsledky programu *Iftop* běžícím na linuxovém serveru (obr. 7.9). Je zde znázorněno zatížení síťového rozhraní během provádění testu. Můžeme určit např. zdrojovou a cílovou adresu probíhajícího spojení nebo aktuální přenosovou rychlost v příchozím i odchozím směru. Šedá barva pozadí u daných spojení graficky znázorňuje celkovou přenosovou rychlost pro logaritmické měřítko v horní části obrazovky.



Obr. 7.5: Graf závislosti počtu transakcí za sekundu na uplynulém čase.



Obr. 7.6: Sumarizované výsledky: procento úspěšnosti posledního HTTP testu.

Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	10804	180	Minimum	120.0	120.0	0.12	11.376	0.0	Attempted	10804
	Successful	5459	90	Maximum	29896.0	29896.0	13972.901	21931.088	21921.703	Established	7526
	Unsuccessful	5345	89	Average	2910.0	2910.0	2331.598	2236.376	1230.98		
	Aborted	0	0								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Test_0001_0	100.000%	0.0	0 % after 0 seconds	HTTP 1.1	Enabled	50	2

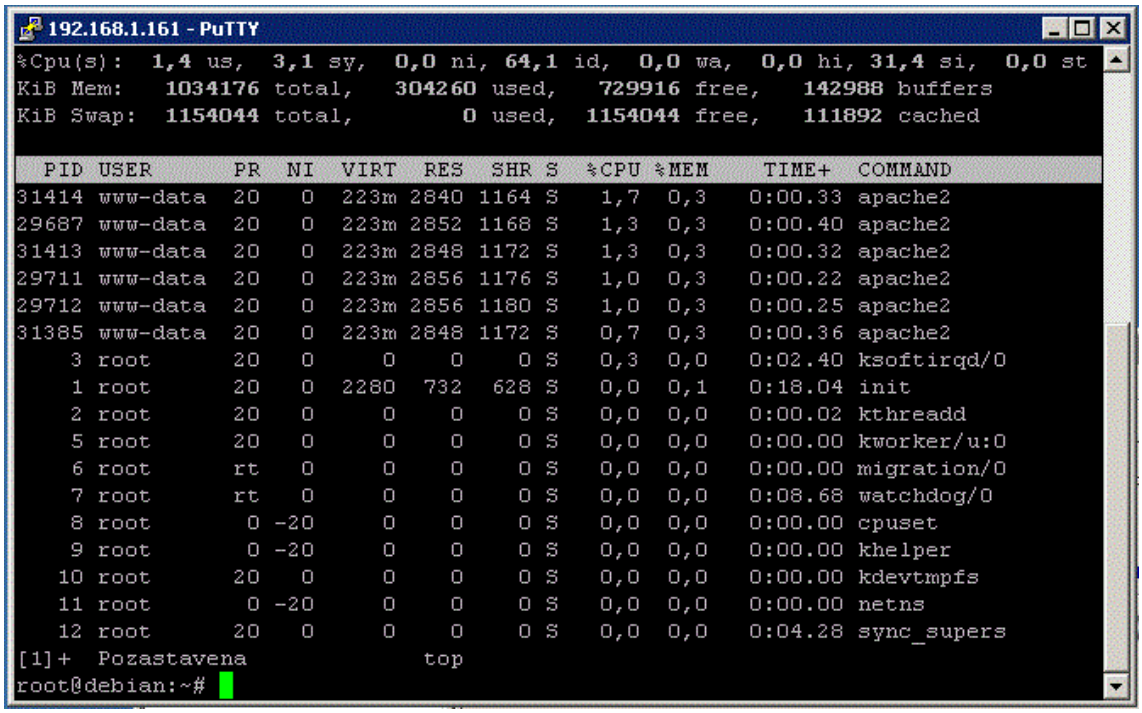
Transaction Summary	Test	Count	Transactions (Sub-Commands included)							
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted
	Test_0001_0	1	90	10804	5459	5345	0	50.52	49.47	0.0
	Totals	1		10804	5459	5345	0	50.52	49.47	0.0

Obr. 7.7: Sumarizované výsledky pro úspěšnost 50,5 %.

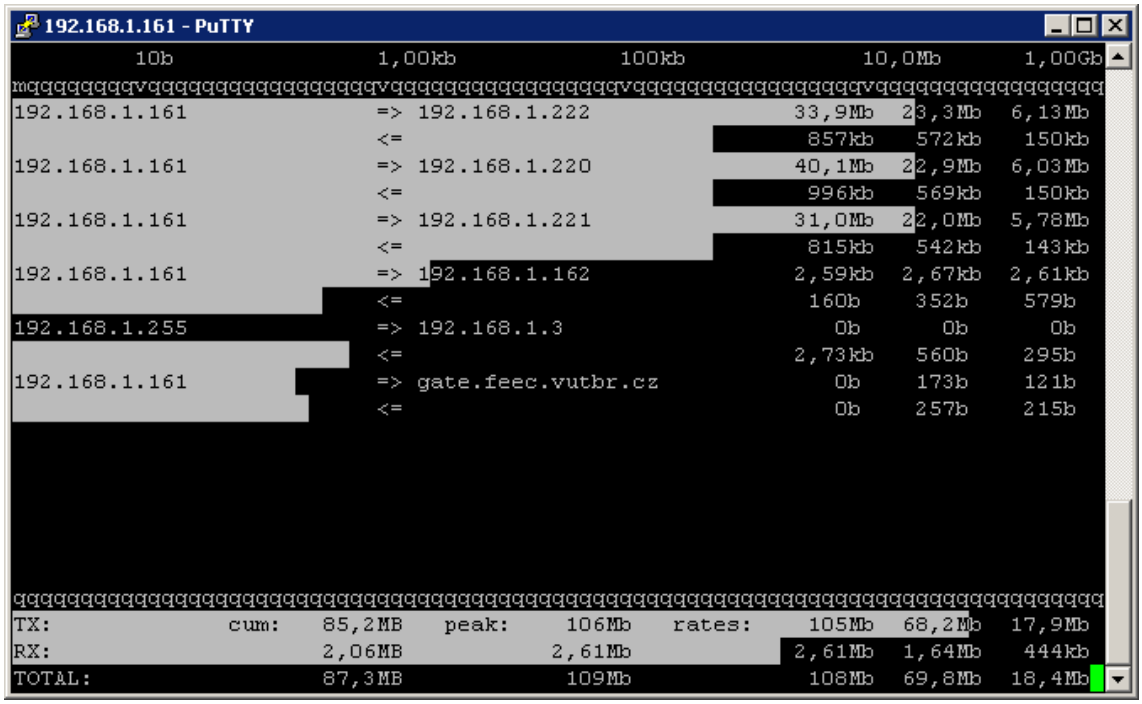
7.1.6 Výsledné zhodnocení

Srovnání výsledků při jaké zátěži bude HTTP server dosahovat dané úspěšnosti je uvedeno v tabulce 7.1. Můžeme si všimnout nepřímé lineární úměry mezi počtem transakcí a úspěšnosti serveru. Nárůst o 30 - 40 transakcí je promítnut do poklesu úspěšnosti o 10 %.

V případě úspěšnosti provedených transakcí 50 % bychom čekali poměrně vysoké vytížení procesoru. Na obrázku 7.8 ovšem nic takového nesledujeme. Z vytížení síťového rozhraní serveru (obr. 7.9) je ale patrno, že je zde dosahováno limitů datového připojení, jelikož server je připojen přes síťový port Fast Ethernet, jehož teoretická



Obr. 7.8: Vytížení serveru při úspěšnosti testu 50,5 %.



Obr. 7.9: Zatížení síťového rozhraní při úspěšnosti testu 50,5 %.

přenosová rychlost je 100 Mb/s. Při plném zatížení serveru (v čase 30 sekund a dále) je na něj kladeno 290 žádostí za sekundu o stažení stránky s velikostí 78,6 kB, čímž

Tab. 7.1: Srovnání množství zátěže s výslednou úspěšností.

Množství zátěže	Úspěšnost testu	Transakce	Úspěšné	Neúspěšné
150 transakcí/s	90,6 %	5 370	4 866	504
190 transakcí/s	79,2 %	6 766	5 357	1 409
220 transakcí/s	69,1 %	7 873	5 437	2 436
250 transakcí/s	57,9 %	9 434	5 462	3 972
290 transakcí/s	50,5 %	10 804	5 459	5 345

se dostáváme na přenosovou rychlost 162 Mb/s (20,3 MB/s) potřebnou pro splnění kladených žádosti.

Z uvedených skutečností vyplývá, že úzkým hrdlem naší sítě je přenosová cesta (rychlost síťového rozhraní), která omezuje množství průchozích transakcí. Tímto jsme vytvořili ukázkový příklad toho, jak může odepření služby vypadat v běžné praxi, aniž by se jednalo o cílený útok. Stačí aby se během krátké doby přihlásilo enormní množství uživatelů (např. slevová akce pro prvních 100 klientů) a v případě, že na to není daný server se svým připojením dimenzován, nastane vysoká míra neúspěšných transakcí nebo v horším případě může dojít k úplnému odepření služby.

7.2 FTP test úspěšnosti

Cíl testu je obdobný jako u HTTP. To znamená, že budeme hledat velikost zátěže, pro kterou bude FTP server postupně vykazovat klesající úspěšnost od 90 % až po 50 %. V záložce *Client* tedy nastavíme:

- *Actions*: `ftp://192.168.1.161/text_file.txt`
- *IP Adress Range*: 192.168.1.220-192.168.1.240
- *Load Specification*: SimUsers/second
- *Test Duration*: 60
- *Time Scale*: Seconds
- *Ports*: 192.168.1.2:8,8

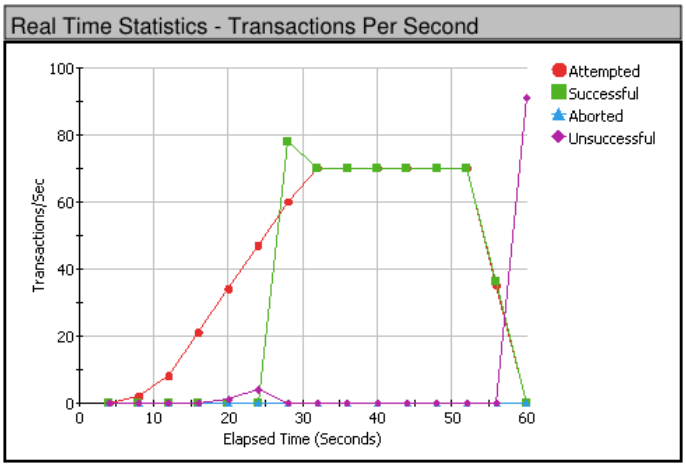
a zátěž, položku *Maximum Load Height*, budeme v jednotlivých testech postupně navyšovat. Velikost stahovaného souboru `text_file.txt` je 80,3 kB, což je přibližně stejně jako u velikosti webové stránky u HTTP testu.

7.2.1 Výsledky pro úspěšnost 90 % a 85 %

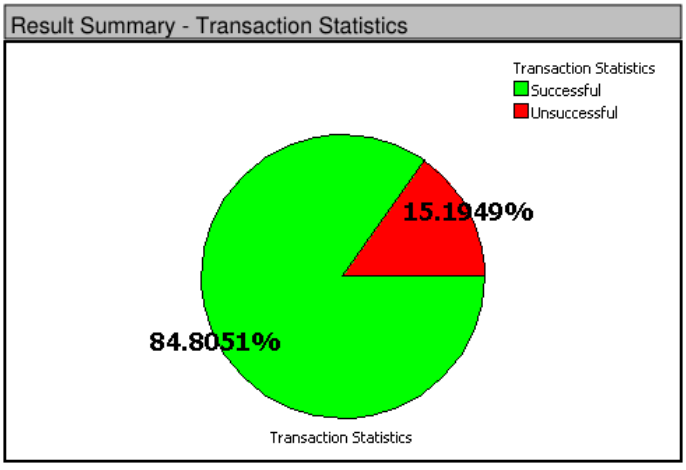
Při nastavení *Maximum Load Height* na počet 9, tzn. **9 simulovaných uživatelů za sekundu**, byla výsledná úspěšnost 90 %. Celkem se provedlo 367 transakcí, z toho

330 úspěšných a 37 neúspěšných.

Další postupné zvyšování zátěže až na **70 simulovaných uživatelů za sekundu** vykazovalo stále stejnou úspěšnost. Konkrétně 85 %. Máme zde tedy poměrně široký rozsah zátěže, který je zpracováván FTP serverem s konstantní úspěšnosti. Pro 70 simulovaných uživatelů za sekundu jsme zaznamenali celkem 2 514 transakcí, úspěšných z toho bylo 2 132 a neúspěšných 382. Na obrázku 7.10 můžeme vidět počet cekových, úspěšných, neúspěšných a předčasně ukončených transakcí za sekundu v průběhu doby trvání testu. Graf procentuální úspěšnosti a sumarizované výsledky jsou na obrázcích 7.11 a 7.12. V logovacím souboru FTP serveru



Obr. 7.10: Graf závislosti počtu transakcí za sekundu na uplynulém čase.



Obr. 7.11: Sumarizované výsledky: procento úspěšnosti FTP testu.

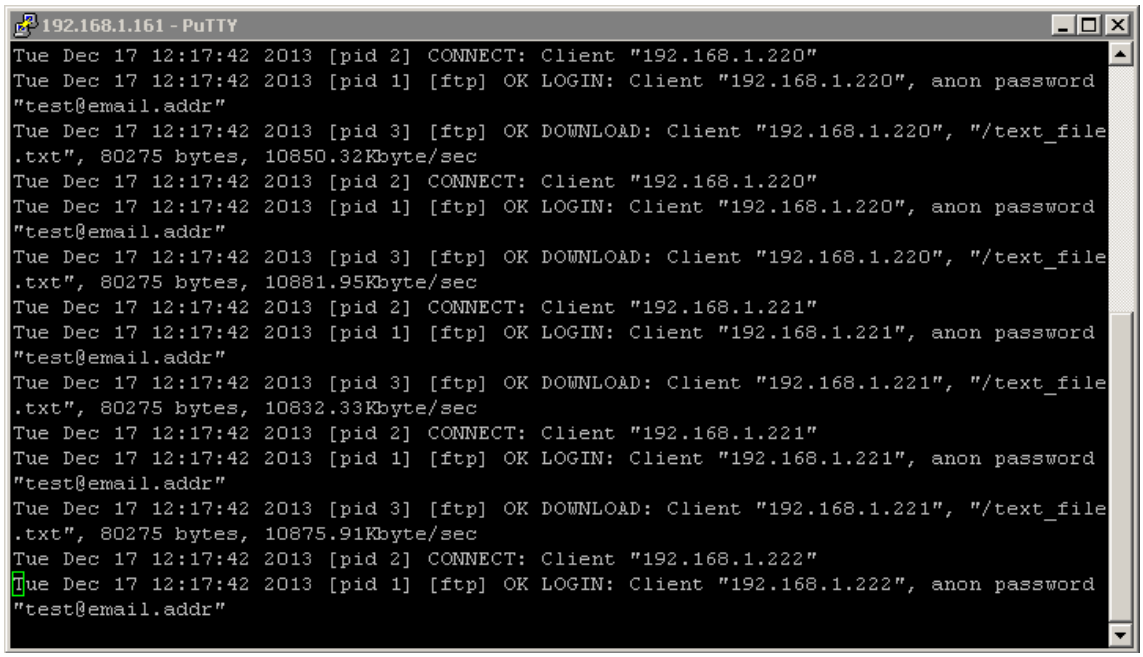
(`/var/log/vsftpd.log`) najdeme informace o činností jednotlivých generovaných uživatelů (např. datum a čas připojování, přihlašování a stahování dat). Část výpisu z logovacího souboru je na obrázu 7.13.

Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	2514	-	Minimum	0.0	17.0	0.084	0.0	0.0	Attempted	4646
	Successful	2132	-	Maximum	0.0	14397.0	13963.751	0.0	421.472	Established	4628
	Unsuccessful	382	-	Average	NaN	548.0	174.679	0.0	8.611		
	Aborted	0	-								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Test_0001_0	100.000%	0.0	0 % after 0 seconds	FTP	Enabled	50	2

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted
	Test_0001_0	1	35	2514	2132	382	0	84.8	15.19	0.0
	Totals	1		2514	2132	382	0	84.8	15.19	0.0

Obr. 7.12: Sumarizované výsledky pro úspěšnost 85 %.

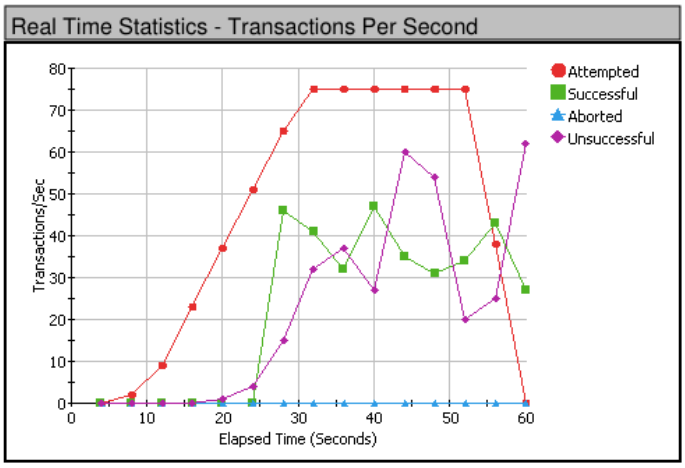


Obr. 7.13: Výpis z logovacího souboru FTP serveru.

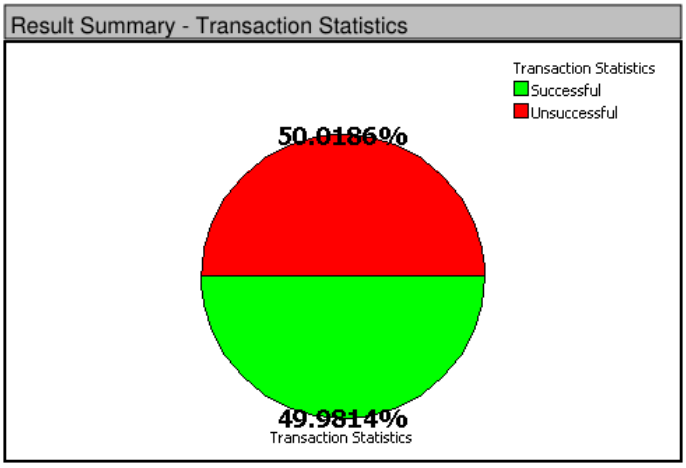
7.2.2 Výsledky pro úspěšnost 50 %

Pouhý nárůst zátěže o 5 simulovaných uživatelů za sekundu znamenal prudký pokles úspěšnosti o 35 %. Oproti předchozímu testu bylo pro **75 simulovaných uživatelů za sekundu** vygenerováno pouze o 175 transakcí více, celkově tedy 2 689. Polovina z nich byla neúspěšná, tj. 1 345 transakcí. Počet generovaných transakcí za sekundu

během trvání testu vyčteme z grafu na obrázku 7.14. Graf procentuální úspěšnosti a sumarizované výsledky jsou na obrázcích 7.15 a 7.16.



Obr. 7.14: Graf závislosti počtu transakcí za sekundu na uplynulém čase.



Obr. 7.15: Sumarizované výsledky: procento úspěšnosti FTP testu.

Pro srovnání s HTTP testem bylo během zátěžového testu také zaznamenáno vytížení síťového rozhraní programem *Iftop* (obr. 7.17).

7.2.3 Výsledné zhodnocení

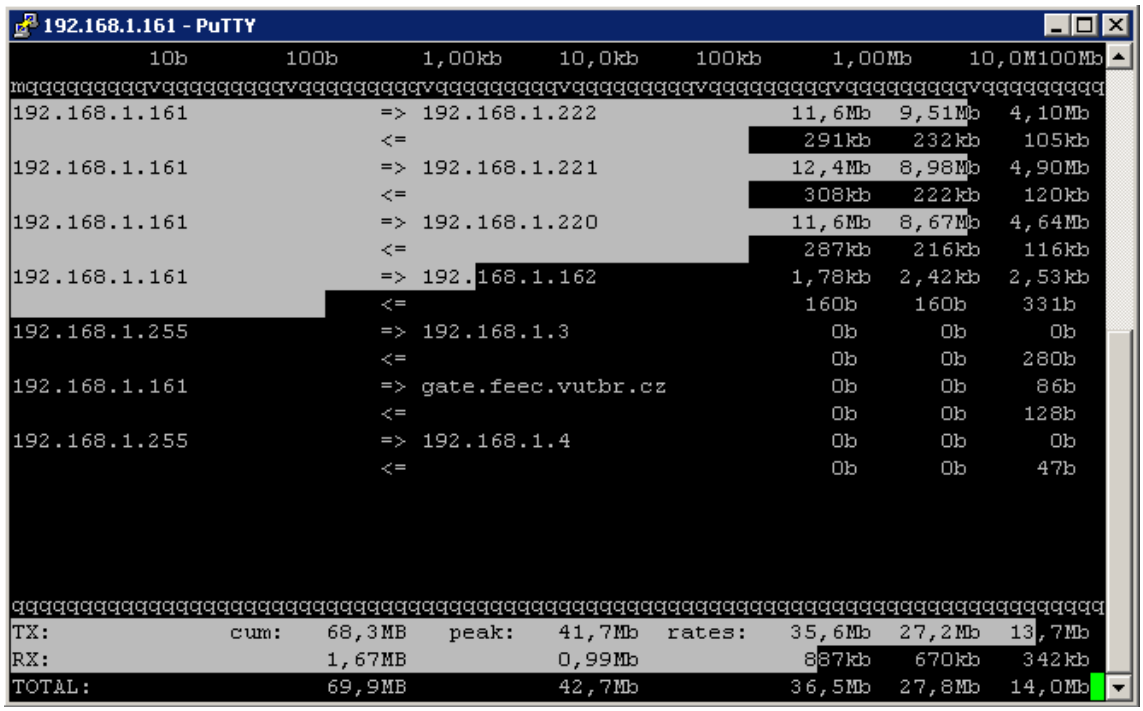
Již při prvních zátěžových testech FTP serveru bylo zřejmé, že výsledky budou mít zcela jiný charakter než výsledky zátěže serveru HTTP. Zatímco u testu HTTP serveru jsme sledovali lineární pokles úspěšnosti vzhledem k lineárnímu nárůstu zátěže, u FTP serveru jsme zaznamenali rapidní skok z úspěšnosti 85 % na 50 % při nepatrném zvýšení zátěže o 5 uživatelů (z 70 na 75).

Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	2689	-	Minimum	0.0	292.0	0.071	0.0	0.0	Attempted	4033
	Successful	1344	-	Maximum	0.0	21087.0	13999.525	0.0	14798.702	Established	3839
	Unsuccessful	1345	-	Average	NaN	4132.0	847.749	0.0	506.241		
	Aborted	0	-								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Test_0001_0	100.000%	0.0	0 % after 0 seconds	FTP	Enabled	50	2

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted
	Test_0001_0	1	22	2689	1344	1345	0	49.98	50.01	0.0
	Totals	1		2689	1344	1345	0	49.98	50.01	0.0

Obr. 7.16: Sumarizované výsledky pro úspěšnost 50 %.



Obr. 7.17: Vytížení síťové karty při úspěšnosti 50 %.

V našem případě bychom možná opět očekávali, že důvodem nízké úspěšnosti je nedostatečná přenosový rychlost spojení, či maximální vytíženost síťového rozhraní. Na obrázku 7.17 ovšem vidíme, že celková přenosová rychlost všech datových toků nedosahuje ani 50 Mb/s. Z toho vyplývá, že schopnost síťového rozhraní není v tomto

případě důvodem nízké úspěšnosti FTP serveru. Závěrem je tedy skutečnost, že při zátěži 75 simulovaných uživatelů za sekundu je FTP server z důvodu svého vytížení schopen pracovat pouze s padesáti procentní úspěšností.

7.3 FTP test (různé velikosti souborů)

Cílem testu je ověřit vliv velikosti souboru na odezvu serveru.

Z důvodu připojení serveru přes port Fast Ethernet je potřeba před samotným testováním určit velikost zátěže tak, aby i při největším testovaném souboru nebyla potřebná přenosová rychlost větší než 100 Mb/s. V případě, že by tato podmínka nebyla splněna, zanášela by se do výsledků chyba způsobena omezenou přenosovou rychlostí síťového spojení. Velikost největšího testovaného souboru je 3,51 MB. Abychom nepřesáhli zmíněnou přenosovou rychlost, budeme generovat 3 simulované uživatele za sekundu. V záložce *Client* tedy nastavíme:

- *IP Adress Range*: 192.168.1.220-192.168.1.240
- *Maximum Load Height*: 3
- *Load Specification*: SimUsers/second
- *Test Duration*: 60
- *Time Scale*: Seconds
- *Ports*: 192.168.1.2:8,8

Toto nastavení bude pro všechny zátěžové testy stejné. Měnit budeme pouze záznam v bloku *Actions*, kde budeme zadávat požadavek na server ke stažení souboru přes FTP protokol `ftp://192.168.1.161/požadovaný_soubor.přípona`. Celkem budeme provádět 3 zátěžové testy pro 3 soubory různých velikostí:

- `text_file.txt` – textový soubor o velikosti 80,3 kB
- `picture_file.jpg` – obrázek o velikosti 1,53 MB
- `mp3_file.mp3` – hudební soubor o velikosti 3,51 MB.

7.3.1 Výsledky pro soubor s 80,3 kB

Pro první testování byl použit textový soubor `text_file.txt` s nejmenší velikostí. Během testu se přeneslo celkem 8,7 MB FTP dat. Úspěšnost testu byla 97,37 %. Proběhlo 114 transakcí, z toho pouze 3 neúspěšné (obr. 7.18). Průměrná doba odezvy serveru byla 1 603,7 ms. V tabulce FTP výsledků (obr. 7.19) najdeme také průměrné časy kontrolního připojení (1 604,2 ms), přihlášení (18,1 ms) a datového připojení (57,9 ms).

Celkový počet zřízených TCP spojení (222) a průměrný čas potvrzení požadavku na TCP spojení (586,6 ms) najdeme na obrázku 7.20.

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted
	Test_0001_0	1	2	114	111	3	0	97.36	2.63	0.0
	Totals	1		114	111	3	0	97.36	2.63	0.0

Response Time (ms)		
Minimum	Maximum	Average
16.0	14000.0	1603.728

Obr. 7.18: Sumarizované výsledky pro soubor s 80,3 kB

FTP Summary	
Total Number Of Attempted FTP Sessions	114
Total Number Of Successful FTP Control Sessions	111
Total Number Of Successful FTP Data Sessions	111
Total Number Of Unsuccessful FTP Control Sessions	3
Total Number Of Aborted FTP Control Sessions	0
Total Number Of FTP Data Transfered (KBytes)	8701.685
Average FTP Control Connection Time (ms)	1604.222
Average FTP Login Time (ms)	18.113
Average FTP Data Connection Time (ms)	57.942

Obr. 7.19: FTP výsledky pro soubor s 80,3 kB

Tcp Summary	
Total Attempted TCP Connections	225
Total Established TCP Connections	222
Minimum Time To TCP SYN/ACK (ms)	0.121
Maximum Time To TCP SYN/ACK (ms)	13644.227
Average Time To TCP SYN/ACK (ms)	586.551

Obr. 7.20: TCP výsledky pro soubor s 80,3 kB.

7.3.2 Výsledky pro soubor s 1,53 MB

Pro druhé testování byl použit JPEG obrázek `picture_file.jpg`. FTP dat bylo během testu přeneseno celkem 165,7 MB. Úspěšnost testu byla 97,37 %. Proběhlo 114 transakcí, z toho 111 úspěšných a 3 neúspěšné (obr. 7.21). Průměrná doba odezvy serveru byla 2 372,1 ms. V tabulce sumarizovaných FTP výsledků (obr. 7.22) jsou

průměrné časy kontrolního připojení (2 372,5 ms), přihlášení (20,8 ms) a datového připojení (819,7 ms). Celkový počet zřízených TCP spojení (222) a průměrný čas potvrzení požadavků na TCP spojení (586,6 ms) najdeme na obrázku 7.23.

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted
	Test_0001_0	1	2	114	111	3	0	97.36	2.63	0.0
	Totals	1		114	111	3	0	97.36	2.63	0.0

Response Time (ms)		
Minimum	Maximum	Average
139.0	17025.0	2372.053

Obr. 7.21: Sumarizované výsledky pro soubor s 1,53 MB.

FTP Summary	
Total Number Of Attempted FTP Sessions	114
Total Number Of Successful FTP Control Sessions	111
Total Number Of Successful FTP Data Sessions	111
Total Number Of Unsuccessful FTP Control Sessions	3
Total Number Of Aborted FTP Control Sessions	0
Total Number Of FTP Data Transfered (KBytes)	165749.991
Average FTP Control Connection Time (ms)	2372.542
Average FTP Login Time (ms)	20.785
Average FTP Data Connection Time (ms)	819.711

Obr. 7.22: FTP výsledky pro soubor s 1,53 MB.

Tcp Summary	
Total Attempted TCP Connections	225
Total Established TCP Connections	222
Minimum Time To TCP SYN/ACK (ms)	0.118
Maximum Time To TCP SYN/ACK (ms)	13644.257
Average Time To TCP SYN/ACK (ms)	591.964

Obr. 7.23: TCP výsledky pro soubor s 1,53 MB.

7.3.3 Výsledky pro soubor s 3,51 MB

Posledním testovaným souborem byl hudební soubor `mp3_file.mp3`. Přeneseno bylo během testu celkem 380,5 MB FTP dat. Úspěšnost testu byla opět 97,37 %. Celkem proběhlo 114 transakcí. Úspěšných bylo 111 a neúspěšné byly 3 (obr. 7.24). Průměrná doba odezvy serveru byla 9 045,2 ms. V tabulce sumarizovaných FTP výsledků (obr. 7.25) jsou průměrné časy kontrolního připojení (9 045,7 ms), přihlášení (62,5 ms) a datového připojení (7 442,3 ms). Na obrázku 7.26 najdeme celkový počet zřízených TCP spojení (222) a průměrný čas potvrzení požadavku na TCP spojení (670,3 ms).

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted
	Test_0001_0	1	1	114	111	3	0	97.36	2.63	0.0
	Totals	1		114	111	3	0	97.36	2.63	0.0

Response Time (ms)		
Minimum	Maximum	Average
2621.0	39271.0	9045.246

Obr. 7.24: Sumarizované výsledky pro soubor s 3,51 MB

FTP Summary	
Total Number Of Attempted FTP Sessions	114
Total Number Of Successful FTP Control Sessions	111
Total Number Of Successful FTP Data Sessions	111
Total Number Of Unsuccessful FTP Control Sessions	3
Total Number Of Aborted FTP Control Sessions	0
Total Number Of FTP Data Transfered (KBytes)	380546.373
Average FTP Control Connection Time (ms)	9045.711
Average FTP Login Time (ms)	62.480
Average FTP Data Connection Time (ms)	7442.250

Obr. 7.25: FTP výsledky pro soubor s 3,51 MB.

7.3.4 Výsledné zhodnocení

Z výše provedených testů vidíme, že velikost testovaných souborů neměla žádný vliv na výslednou úspěšnost FTP serveru. Všechny testy proběhly s 97,37 procentní

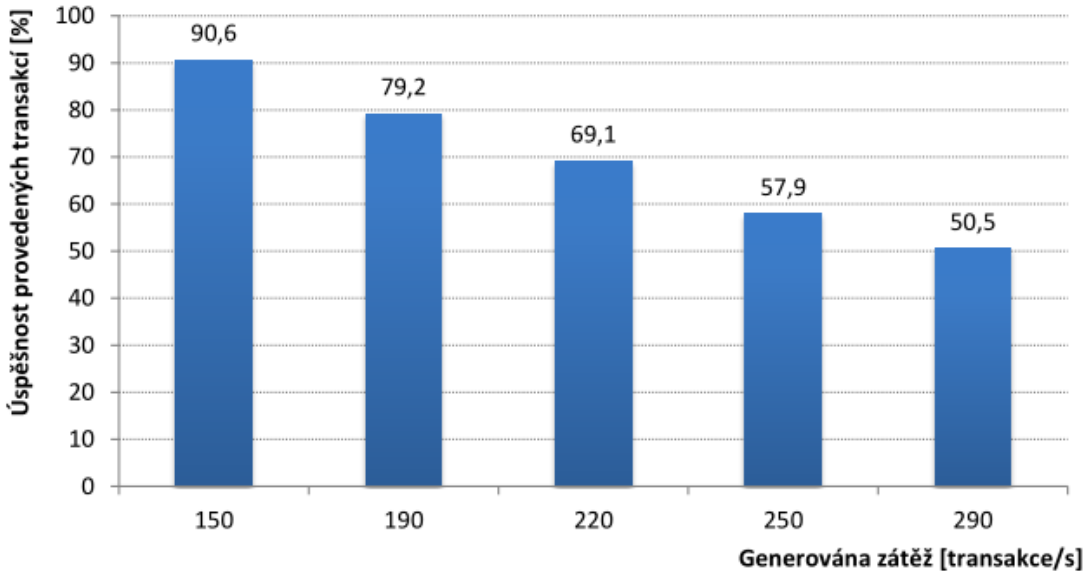
Tcp Summary	
Total Attempted TCP Connections	225
Total Established TCP Connections	222
Minimum Time To TCP SYN/ACK (ms)	0.105
Maximum Time To TCP SYN/ACK (ms)	13644.299
Average Time To TCP SYN/ACK (ms)	670.333

Obr. 7.26: TCP výsledky pro soubor s 3,51 MB.

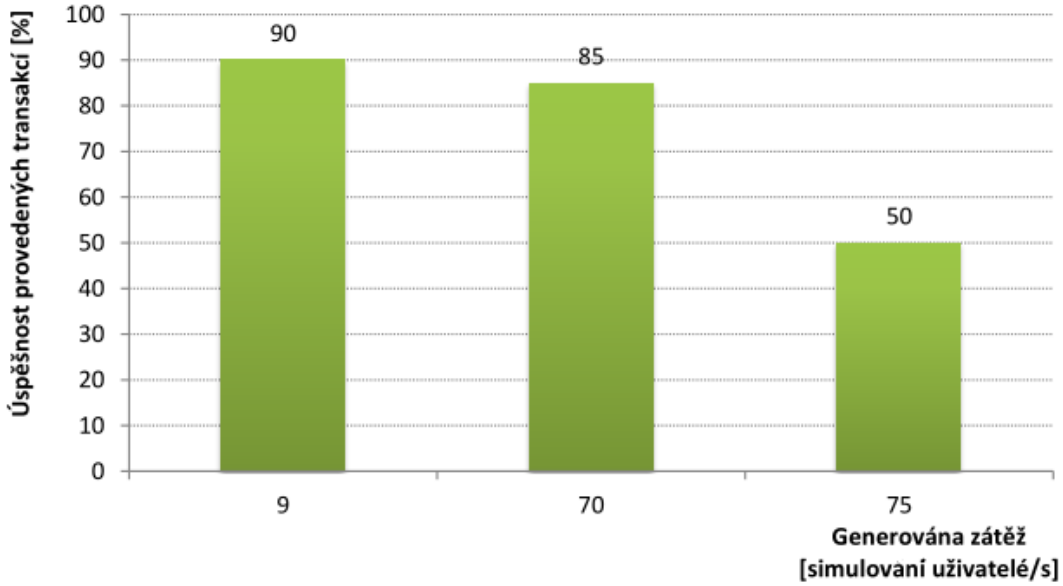
úspěšnosti. Proběhlo také stejné množství zřízených TCP spojení (222). Z výsledků je zřejmé, že s rostoucí velikostí souboru roste průměrná doba odezvy serveru. Konkrétně z 1 603,7 ms pro nejmenší soubor na 9 045,2 ms pro největší soubor. Narůstá tím i doba potřebná pro FTP přihlášení (z 18,1 ms až na 62,5 ms). Ve výsledcích pro TCP komunikaci (obr. 7.20, 7.23, 7.26) si můžeme všimnout nárustu průměrného času potřebného pro potvrzení požadavku na TCP spojení (SYN/ACK) vzhledem k rostoucí velikosti souboru. Průměrný čas vzrostl z 586,6 ms pro nejmenší soubor na 670,3 ms pro největší soubor. Závěrem tedy můžeme říci, že rostoucí velikost souboru s sebou přináší delší odezvu serveru a prodloužení doby kontrolních sekvencí daných protokolů.

7.4 Přehled výsledků zátěžových testů

Výsledky úspěšnosti provedených transakcí zátěžových testů v závislosti na množství generované zátěže jsou přehledně zobrazeny v grafu pro HTTP testy (obr. 7.27) a v grafu pro FTP testy (obr. 7.28).



Obr. 7.27: Graf úspěšnosti HTTP testů.



Obr. 7.28: Graf úspěšnosti FTP testů.

8 ZÁTĚŽOVÉ TESTOVÁNÍ S DDOS ÚTOKY

V rámci této kapitoly se zaměříme na zátěžové testování a měření vlivu DDoS útoků na webové služby poskytované serverem. Konkrétně budou realizovány útoky SynFlood, UDPFlood, XMasTree, ResetFlood a ARPFlood. Z výsledků testování bude následně vybrán nejúčinnější DDoS útok, proti kterému budou v následující kapitole 9 navrženy ochranné prostředky na úrovni „end-host“ ochrany, čili ochrany implementované na koncovém zařízení. Ochranné prostředky budou poté zavedeny na testovaný server a podrobeny vybranému DDoS útoku. Na základě získaných výsledků bude provedeno zhodnocení účinnosti jednotlivých ochran.

8.1 Testování „výkonnosti“ serveru

Cílem této sady testů je získání „křivky úspěšnosti“ serveru. Výsledkem bude závislost úspěšnosti serveru na počtu uskutečněných spojení za sekundu. Z této charakteristiky bude poté vybrána hodnota počtu spojení za sekundu, která bude sloužit jako referenční hodnota zátěže pro realizaci následných DDoS útoků. Zátěžové testy budou zaměřeny na webové služby poskytované serverem.

8.1.1 Nastavení testů

K získání požadovaných výsledků nám postačí parametry profilu „Quick test“ (viz část 3.1.1). V záložce *Client* tedy nastavíme:

- *Actions*: `get http://10.0.0.2/index.php`
- *IP Adress Range*: 10.0.0.3-10.0.0.150
- *Load Specification*: Connections/second
- *Test Duration*: 60
- *Time Scale*: Seconds
- *Ports*: 192.168.1.2:0,0

a zátěž, položku *Maximum Load Height*, budeme v jednotlivých testech postupně navyšovat.

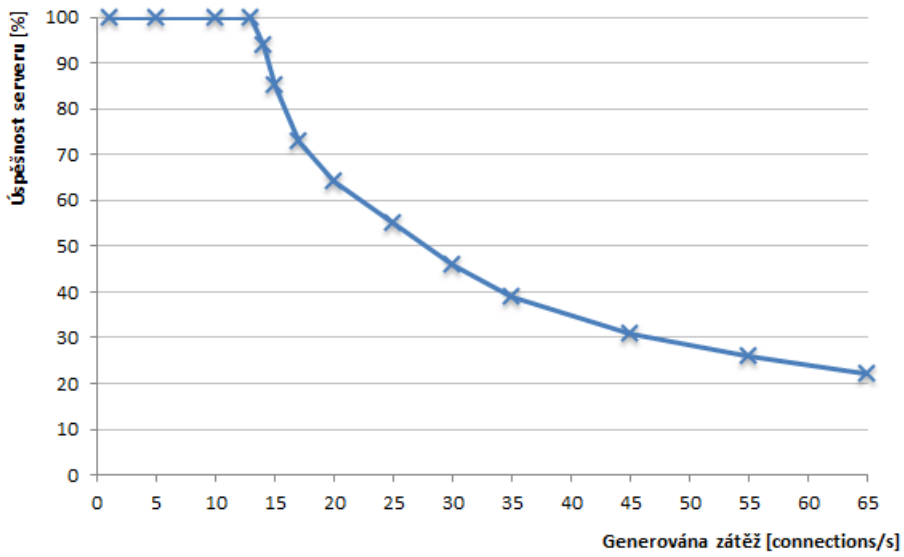
8.1.2 Výsledky testů

Přehled výsledků testů je uveden v tabulce 8.1. Z uvedené tabulky byla vytvořena grafická závislost úspěšnosti serveru na počtu uskutečněných spojení za sekundu (obr. 8.1). Z výsledků je patré, že server dokáže zpracovat zátěž do velikosti 13 spojení za sekundu při zachování 100% úspěšnosti.

Jako referenční hodnotu zátěže pro další testování zvolíme 5 spojení za sekundu, což je hodnota, u které můžeme s jistotu konstatovat stoprocentní úspěšnost serveru (včetně určité rezervy).

Tab. 8.1: Výsledky jednotlivých zátěžových testů.

Množství zátěže	Úspěšnost testu	Celkem spojení	Úspěšné	Neúspěšné
1 connections/s	100 %	39	39	0
5 connections/s	100 %	189	189	0
10 connections/s	100 %	414	414	0
13 connections/s	100 %	495	495	0
14 connections/s	94 %	542	509	33
15 connections/s	85 %	589	500	89
17 connections/s	73 %	622	453	169
20 connections/s	64 %	765	490	275
25 connections/s	55 %	940	519	421
30 connections/s	46 %	1 115	510	605
35 connections/s	39 %	1 290	505	785
45 connections/s	31 %	1 639	509	1 130
55 connections/s	26 %	1 989	518	1 471
65 connections/s	22 %	2 386	525	1 998



Obr. 8.1: Graf úspěšnosti serveru.

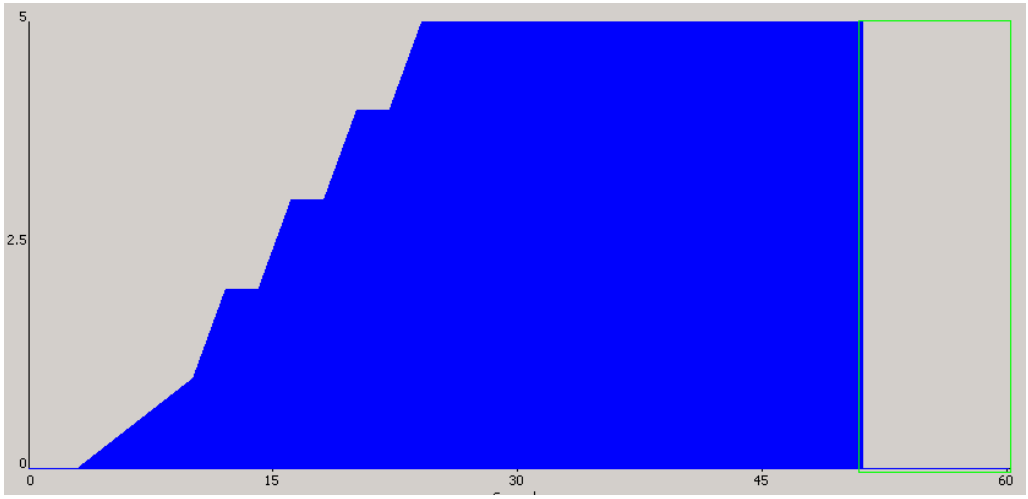
8.2 DDoS útoky

Na základě předchozího měření byla stanovena referenční hodnota zátěže na 5 spojení za sekundu. Tato hodnota bude nastavena jako výchozí pro další testování a bude označovat velikost legitimního datového provozu. K tomuto legitimnímu provozu bude následně přidán datový tok DDoS útoku. Jak již bylo zmíněno v úvodu kapitoly, bude se jednat o útoky SynFlood, UDPFlood, XMasTree, ResetFlood a ARPFlood. Výsledkem bude srovnání vlivu jednotlivých DDoS útoků na webové služby poskytované serverem.

K měření vlivu DDoS útoků již budeme potřebovat parametry profilu „Advanced test“, který nám oproti profilu „Quick test“ nabízí podrobnější nastavení jednotlivých parametrů testů a především možnost realizace DDoS útoků. V programu *TestCenter Layer 4-7 Application* tedy vytvoříme nový test, jehož nastavení bude vypadat následovně:

- *Actions*: `get http://10.0.0.2/index.php`
- *IP Adress Range*: 10.0.0.3/16-10.0.10.150/16
- *Load Specification*: Connections/second
- *Total Duration*: 60
- *Time Scale*: Seconds
- *Ports*: 192.168.1.2:0,0

Na obrázku 8.2 je graf průběhu testu zobrazující nárust zátěže (legitimního provozu) v závislosti na čase.



Obr. 8.2: Graf nárustu zátěže v závislosti na čase.

Globální nastavení DDoS útoků, společné pro všechny následné testy, je uvedeno v tabulce 8.2. V každém testu se dále nastavuje počet generovaných DDoS paketů za sekundu (*PacketRate*) a celkové množství DDoS paketů (*PacketsToGenerate*).

Tab. 8.2: Globální nastavení DDoS útoků.

Parametr	Hodnota
StartingSourceMACAddress	05:00:00:00:00:01
MACAddressIncrement	+1
StartingDestMACAddress	00:15:17:14:18:83
StartingSourceIPAddress	10.1.0.3
StartingDestIPAddress	10.0.0.2
GlobalStartDelay	20 000 ms
RepeatCount	28

Tyto dva parametry budou vždy nastaveny na stejnou hodnotu a jejich postupným zvyšováním budeme určovat velikost DDoS zátěže. Výsledkem pak bude graf zobrazující závislost úspěšnosti serveru na velikosti generované DDoS zátěže. Doba po kterou bude DDoS útok aktivní se určuje podle vzorce:

$$t = RepeatCount \cdot \frac{PacketsToGenerate}{PacketRate}[\mathrm{s}]$$

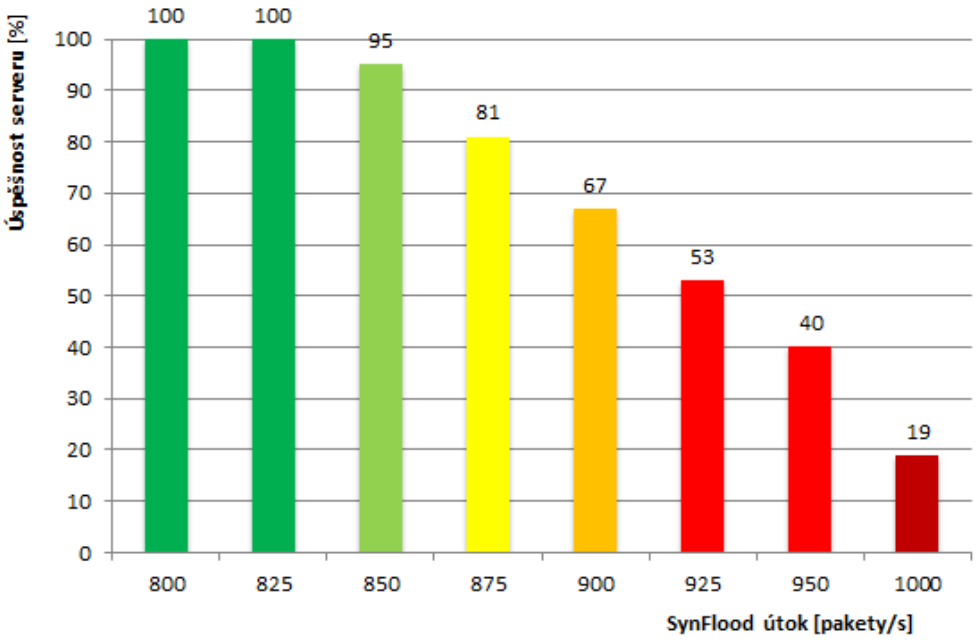
V našem případě dobu DDoS útoků určuje parametr *RepeatCount*, který je nastaven na hodnotu 28. Útok tedy bude trvat 28 sekund a začne se generovat 20 sekund po startu testu (viz tabulka 8.2). V neposlední řadě je potřeba funkci DDoS útoku aktivovat zatrhnutím položky *Enable DDOS Functionality* v záložce *Client > Configure*.

8.2.1 SynFlood

Prvním testovaným DDoS útokem je velice známy útok SynFlood. Sumarizovaný výsledek realizovaných testů ve formě grafu je na obrázku 8.3. Vyjadřuje úspěšnost serveru v závislosti na zvyšující se generované zátěži DDoS útoku (pakety/s). Můžeme si povšimnout, že server dokázal odolávat útoku až do zátěže 825 DDoS paketů/s. Při dalším navyšování generované DDoS zátěže začala úspěšnost serveru rychle klesat. K 50 % úspěšnosti se přiblížil při hodnotě zátěže 925 DDoS paketů/s a při zátěži 1000 DDoS paketů/s byla úspěšnost serveru již pouze 19 %[1]. Průběh poklesu úspěšnosti serveru je přibližně lineární.

Během testování byl průběžně kontrolován stav testovaného serveru přes vzdálené připojení pomocí SSH klienta. V průběhu generování DDoS útoku byl zaznamenán

[1]Úspěšnost serveru 19 % odpovídá úspěšným spojením, které proběhly do doby, než začal být DDoS útok aktivní (prvních 32 spojení). Prakticky to znamená, že od chvíle, kdy byl útok aktivní, nedošlo k žádnému úspěšnému spojení se serverem. Při zátěži 1000 DDoS paketů/s vykazuje tedy útok 100 % účinnost.



Obr. 8.3: Úspěšnost serveru při SynFlood útoku.

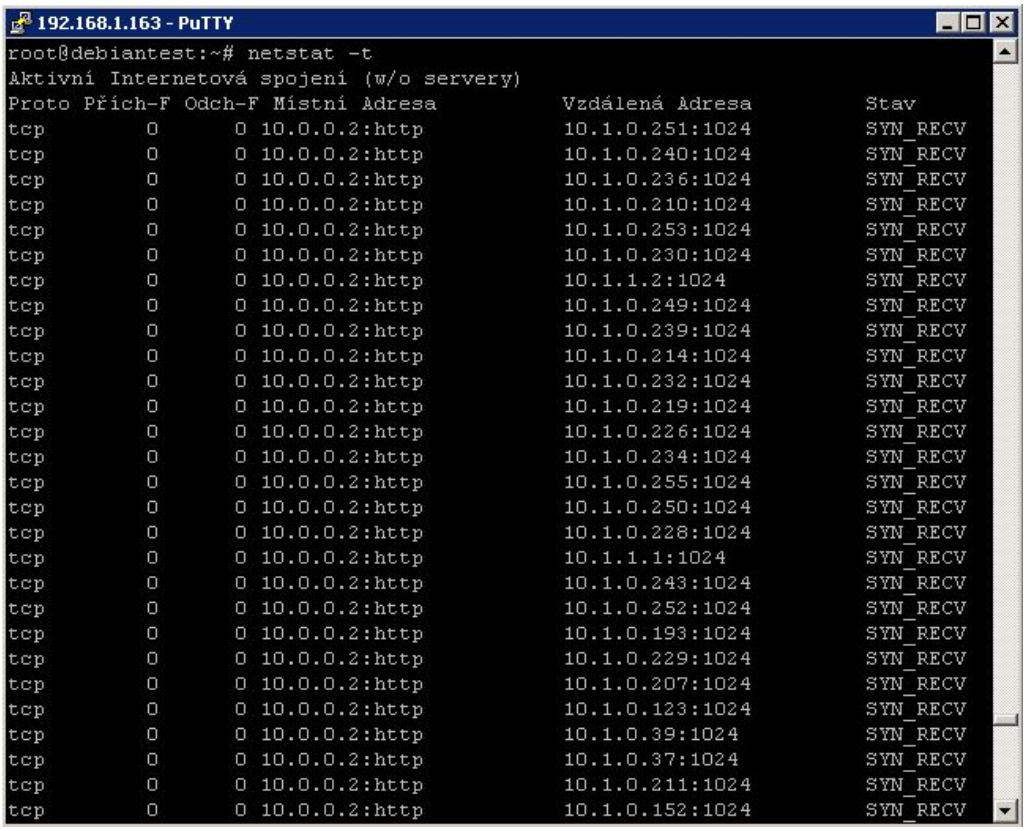
výpis aktivních TCP spojení pomocí příkazu `netstat -t`. Výpis TCP spojení je na obrázku 8.4. Lze z něj vyčíst cílovou IP adresu a port, ke kterému se spojení vztahuje. Dále vzdálenou adresu, čili IP adresu (a port) zdroje spojení, a také stav ve kterém se spojení nachází. Můžeme zde sledovat velkého množství spojení, které jsou ve stavu SYN_RECV. Jedná se o stav, kdy byl přijat serverem SYN paket, na který server odpovídá paketem SYN/ACK a čeká na potvrzení ACK ze strany klienta. Vzhledem k tomu, že SYN paket byl generován z rozsahu adres pro generování SynFlood útoku, server odpověď nedostane a spojení v tomto napůl otevřeném stavu zůstane. To s sebou přináší velké množství rezervovaných prostředků serveru pro každé spojení. Při určitém počtu napůl otevřených spojení, již server není schopen další prostředky rezervovat a nová spojení přijímat, což vede k zablokování legitimního webového provozu.

8.2.2 UDPFlood

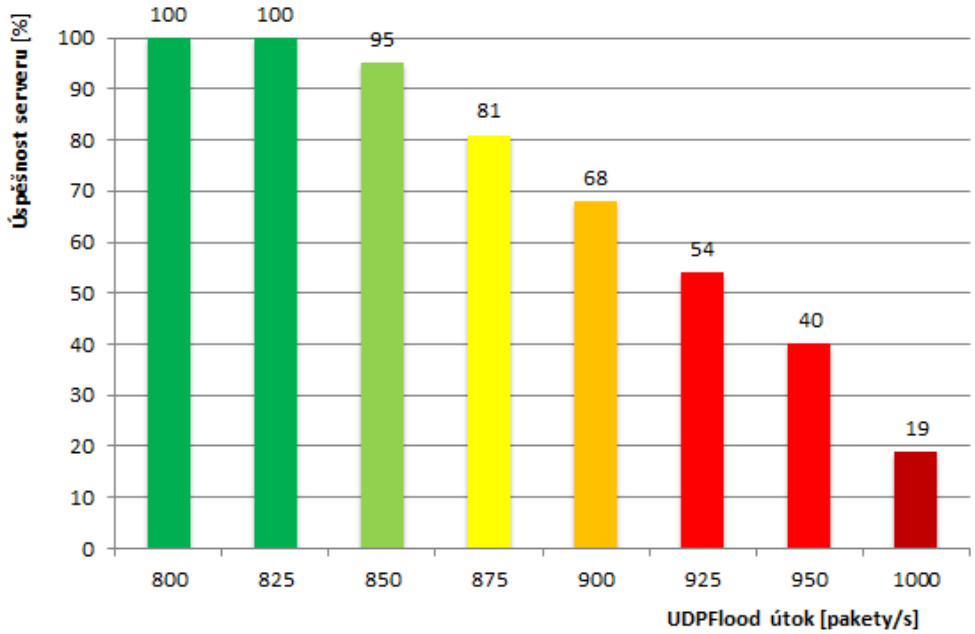
Druhým testovaným DDoS útokem je UDPFlood. Výsledky testování jsou na obrázku 8.5. Srovnáním s výsledky SynFlood útoku (obr. 8.3) si můžeme povšimnout jejich podobného charakteru. U SynFlood útoku je úspěšnost serveru nepatrně nižší.

8.2.3 XmassTree

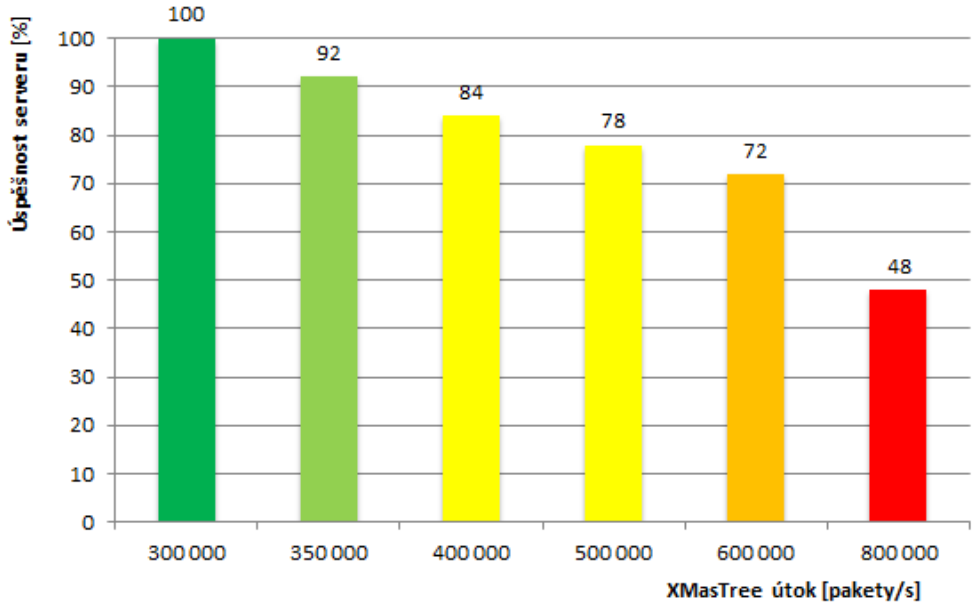
Výsledky z testování jsou zobrazeny na obrázku 8.6. Server dokázal tomuto útoku



Obr. 8.4: Výpis TCP spojení při SynFlood útoku.



Obr. 8.5: Úspěšnost serveru při UDPFlood útoku.



Obr. 8.6: Úspěšnost serveru při XmassTree útoku.

odolávat podstatně lépe než předchozím dvěma útokům. Úspěšnost 100 % si držel až do hodnoty 300 000 DDoS paketů/s. Pokles úspěšnosti byl velice pozvolný. Při dvojnásobné zátěži byla úspěšnost serveru ještě 72 %. Pod hranici 50 % se server dostal až při hodnotě zátěže 800 000 DDoS paketů/s.

Při kontrole stavu testovaného serveru byl zaznamenán vysoký nárůst vytížení procesoru procesem *kworker* a *ksoftirqd*[2]. Na obrázku 8.7 je vidět vytížení procesoru těmito procesy při hodnotě generované zátěže 800 000 DDoS paketů/s. Úspěšnost serveru tedy klesá především z důvodu vysokého vytížení procesoru, který je zaměstnán vysokým počtem přerušení, které způsobují pakety XmassTree útoku.

8.2.4 ResetFlood

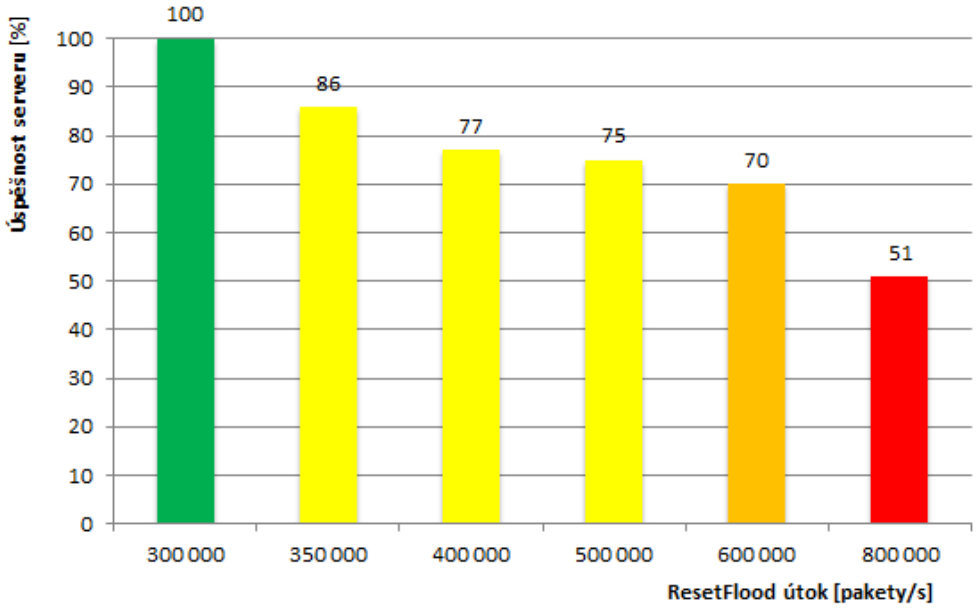
Sumarizovaný výsledek realizovaných útoků ResetFlood je ve formě grafu na obrázku 8.8. Server opět dokázal tomuto útoku odolávat až do zátěže 300 000 DDoS paketů/s. Při navýšení zátěže o 50 000 DDoS paketů/s klesla úspěšnost serveru o 14 % na hodnotu 86 %. Při dalším navyšování zátěže klesala úspěšnost serveru již pozvolněji. K hranici úspěšnosti 50 % se server dostal až při hodnotě zátěže 800 000 DDoS paketů/s.

Během měření bylo stejně jako u předchozího útoku sledováno vytížení procesoru. Opět byl zaznamenán vysoký nárůst vytížení procesoru procesy *kworker* a *ksoftirqd*.

[2]Jedná se o procesy zodpovědné za obsluhu přerušení kontextu procesů, obsluhu vzniklých front, apod.

```
top - 14:31:18 up 1 day, 27 min,  2 users,  load average: 0,54, 0,77, 1,28
Tasks:  78 total,   3 running,  75 sleeping,   0 stopped,   0 zombie
%Cpu(s): 18,8 us,  4,0 sy,  0,0 ni, 26,6 id,  0,5 wa,  0,0 hi, 50,2 si,  0,0 st
KiB Mem:   4060076 total,   466560 used,  3593516 free,    54384 buffers
KiB Swap:  6369276 total,        0 used,  6369276 free,   160096 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
11415 root      20   0     0    0    0 R  49,3  0,0   0:21.66 kworker/0:1
    3 root      20   0     0    0    0 R  49,0  0,0   0:51.15 ksoftirqd/0
11757 www-data  20   0  150m  10m 3284 S  14,6  0,3   0:01.63 apache2
11700 www-data  20   0  150m  10m 3284 S  14,2  0,3   0:06.60 apache2
11730 www-data  20   0  150m  10m 3284 S  13,9  0,3   0:03.10 apache2
 2665 mysql     20   0  455m  56m 7616 S   2,0  1,4   0:58.09 mysqld
 1691 root      20   0 18972  872  616 S   1,0  0,0   0:00.10 rpcbind
11641 root      20   0 22140 1504 1112 R   0,3  0,0   0:01.48 top
11728 root      20   0     0    0    0 S   0,3  0,0   0:00.11 kworker/0:0
    1 root      20   0 10648  812  676 S   0,0  0,0   0:03.09 init
    2 root      20   0     0    0    0 S   0,0  0,0   0:00.00 kthreadd
    6 root      rt   0     0    0    0 S   0,0  0,0   0:00.00 migration/0
    7 root      rt   0     0    0    0 S   0,0  0,0   0:00.48 watchdog/0
    8 root      rt   0     0    0    0 S   0,0  0,0   0:00.01 migration/1
   10 root      20   0     0    0    0 S   0,0  0,0   0:00.12 ksoftirqd/1
   12 root      rt   0     0    0    0 S   0,0  0,0   0:00.10 watchdog/1
   13 root       0 -20     0    0    0 S   0,0  0,0   0:00.00 cpuset
   14 root       0 -20     0    0    0 S   0,0  0,0   0:00.00 khelper
   15 root      20   0     0    0    0 S   0,0  0,0   0:00.00 kdevtmpfs
   16 root       0 -20     0    0    0 S   0,0  0,0   0:00.00 netns
   17 root      20   0     0    0    0 S   0,0  0,0   0:00.16 sync_supers
   18 root      20   0     0    0    0 S   0,0  0,0   0:00.00 bdi-default
   19 root       0 -20     0    0    0 S   0,0  0,0   0:00.00 kintegrityd
   20 root       0 -20     0    0    0 S   0,0  0,0   0:00.03 kblockd
   21 root      20   0     0    0    0 S   0,0  0,0   0:02.15 kworker/1:1
   22 root      20   0     0    0    0 S   0,0  0,0   0:00.03 khungtaskd
```

Obr. 8.7: Vytížení procesoru při XmassTree útoku.



Obr. 8.8: Úspěšnost serveru při ResetFlood útoku.

Procesy jsou zaměstnány obsluhou přerušení vyvolaných TCP pakety s nastaveným příznakem RST (reset), které generuje ResetFlood útok.

8.2.5 ARPFlood

Posledním testovaným DDoS útokem je ARPFlood. Pro realizaci toho útoku je potřeba nakonfigurovat doplňující nastavení. Konkrétně se jedná o nastavení parametrů *ARPHeaderDestEthernetAddress* a *ARPHeaderDestIPAddress*. U prvního z nich je potřeba zadat MAC adresu a u druhého IP adresu testovaného serveru.

Testování probíhalo klasickým způsobem jako u výše zmíněných testů. Ve výsledcích testování ovšem nebyl zaznamenán žádný pokles úspěšnosti vzhledem k rostoucí velikosti DDoS zátěže. Testování probíhalo až do hodnoty zátěže 800 000 DDoS paketů/s, při které byla hodnota úspěšnosti serveru stále 100 %. Při generování zátěže vyšší jak 800 000 DDoS paketů/s již nebyl vygenerován dostatečný počet legitimních spojení. Úspěšnost serveru pro vyšší hodnoty zátěže tedy nebylo možné vyhodnotit.

ARPFlood je označován za útok, který využívá limitů u zařízení jako jsou například přepínače. Ty mají za úkol udržovat v paměti MAC tabulku a řídí jednotlivé ARP zprávy, které jsou přeposílány na fyzické porty zařízeni dle záznamů v MAC tabulce. Jedná se o útok na linkové vrstvě ISO/OSI modelu [22].

Příčina nulového vlivu ARPFlood útoku na úspěšnost serveru může být tedy v tom, že testovaný server není mezilehlé zařízení, ale je koncové zařízení sítě. Tento útok tedy na něj nemá vliv. Pro podporu této teorie bylo provedeno kontrolní měření. Cílem měření bylo otestovat vliv ARPFlood útoku na mezilehlé zařízení, konkrétně na firewall Cisco ASA 5510. Společně s tímto útokem byl změřen i vliv útoku SynFlood. Výsledky kontrolního měření ukázaly značný dopad ARPFlood útoku na úspěšnost firewallu. Účinnost útoku byla dokonce vyšší než u útoku SynFlood. Pro stejnou generovanou zátěž 90 000 DDoS paketů/s byla úspěšnost firewallu při SynFlood útoku 45 %, kdežto při útoku ARPFlood pouhé 3 %.

Na základě získaných výsledků z měření lze s určitou pravděpodobností konstatovat, že ARPFlood je (svým provedením) účinný pouze při útoku na mezilehlá zařízení. Na koncové zařízení vliv nemá.

8.2.6 Vyhodnocení testů s DDoS útoky

Nejúčinnějšími útoky jsou SynFlood a UDPFlood, kterým server dokázal odolávat pouze do zátěže 825 DDoS paketů/s. Skutečnost, že jsou tyto útoky vyhodnoceny jako nejúčinější, odpovídá také současnému stavu v oblasti DDoS útoků, o kterém se pojednává v části 4.2.1. SynFlood útok je v globálním měřítku označován jako nejčastější DDoS útok, který si svou oblibu vysloužil především svou účinností a také

poměrně snadnou realizovatelností. Obdobně je tomu u útoku UDPFlood, který vykazuje přibližně stejnou účinnost. V globáním měřítku sice není zastoupen tak často jako SynFlood útok, ale stále se jedná o jeden z nejčastějších DDoS útoků.

Z výsledků je patrný velký odstup útoků ResetFlood a XmasTree od výše zmíněných. Tyto útoky začínají mít vliv na úspěšnost serveru až od generované zátěže 350 000 DDoS paketů/s. Jedná se o útoky, které nejsou již v současnosti využívány tak často, jako tomu bylo v minulosti. Důvodem je především skutečnost, že velká část zařízení (či systémů) již dokáže těmto útokům do značné míry odolávat, což je prakticky i náš případ. Pakety útoku XmasTree se záměrně špatně nastavenou TCP hlavičkou, které dříve způsobovaly zamrznutí systému, jsou dnes u velké řady zařízení detekovány a úspěšně zahazovány. Je jasné, že s každým neplatným paketem je spojena určitá režie, ale ve srovnání s potřebou rezervace systémových prostředků pro nová otevřená spojení u SynFlood útoku, je režie spojena s neplatnými pakety u útoku XmasTree zanedbatelná. Obdobně je tomu u útoku ResetFlood.

Z měření dále vyplynulo, že ARPFlood útok nemá na úspěšnost serveru prokazatelný vliv.

Během testování bylo také kontrolováno a monitorováno, zda nedochází k zahlcení přenosové trasy mezi serverem a zařízením Spirent Avalanche, či zda není překročená rychlost síťového rozhraní. K zahlcení přenosové trasy ani k překročení datové rychlosti síťového rozhraní během žádného z testů nedošlo.

Na základě získaných výsledků a zhodnocení lze za nejúčinnější DDoS útok označit SynFlood.

9 TESTOVÁNÍ OCHRAN PROTI DDOS ÚTOKŮM

V rámci této kapitoly budou na linuxový server postupně implementovány různé typy „end-host“ ochran proti DDoS útokům. Označení „end-host“ v tomto případě znamená, že ochranné prostředky jsou implementovány na zařízení, které zároveň poskytuje webové a datové služby. Jedná se prakticky o nejlevnější variantu ochrany proti DDoS útokům, jelikož zde není potřeba žádné další zařízení. Při srovnání s jinými typy ochran (fyzické firewally, systémy prevence průniku IPS, apod.) je ale jasné, že se současně jedná o ochranu nejméně účinnou. Cílem této kapitoly je tedy otestovat různé typy „end-host“ ochran proti DDoS útokům, zjistit nakolik jsou účinné a určit, zda má v současnosti smysl „end-host“ ochranu implementovat, či nikoli.

V našem případě se konkrétně zaměříme na ochrany proti útoku SynFlood, který byl v rámci zátěžového testování s DDoS útoky (kap. 8) označen jako nejúčinnější útok.

9.1 Rozdělení jednotlivých ochran

Ochrany proti SynFlood útoku na „end-host“ úrovni jsem pro účely testování rozdělil do zhruba tří oblastí či skupin, přičemž toto rozdělení není v praxi nijak striktní, jelikož se jednotlivé oblasti mohou vzájemně prolínat a doplňovat.

První z nabízených oblastí je ochrana pomocí Apache modulů. Jedná se o způsob ochrany, kdy je k webovému serveru *Apache* doinstalován speciální modul rozšiřující jeho funkci o ochranu před DoS a DDoS útoky.

Druhá oblast ochrany slučuje možnosti úpravy nastavení jádra systému, velikosti TCP/IP zásobníků, časovačů apod., za účelem zvýšení odolnosti systému vůči DDoS útokům. Do této oblasti můžeme zařadit i ochranné funkce systému, které jsou v jádře již implementovány a lze je pomocí úpravy konfiguračních souborů aktivovat či deaktivovat. Jedná se například o funkci systému s názvem *SynCookies*, která má systém chránit proti záplavě paketů, jenž generuje útok SynFlood.

Třetí z nabízených oblastí je ochrana pomocí firewallu[1] implementovaného v jádře linuxového operačního systému. Jedná se o snahu použít a nastavit stavová i nestavová pravidla firewallu tak, aby byl útok těmito pravidly zachycen dříve, než naruší vnitřní funkci systému.

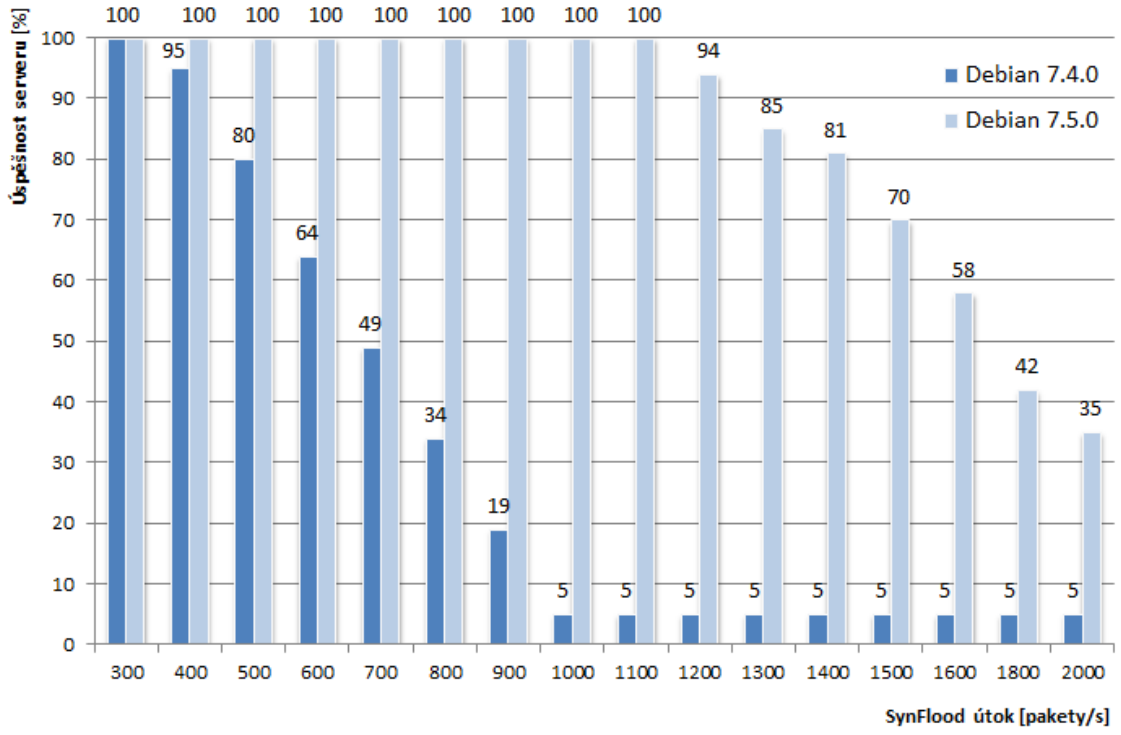
[1] *Iptables*

9.2 Referenční test

Před samotným testováním jednotlivých ochran je potřeba provést referenční měření, vůči kterému se bude určovat jejich účinnost. Nastavení testů bude obdobné jako při testování DDoS útoků v části 8.2. Změna ovšem nastane v délce trvání DDoS útoků, která bude navýšena z 28 sekund na 120 sekund. Celková doba jednotlivých testů pak bude 155 sekund.

Aby byla dodržena bezpečnostní doporučení uvedená v části 4.3, je potřeba před samotnou implementací jednotlivých ochran linuxové jádro aktualizovat. Jedná se o velice důležitý krok, který může přinést větší odolnost vůči DDoS útokům díky novým ochranným mechanizmům, jenž jsou vývojáři do nového jádra implementovány. Byla tedy provedena aktualizace na nejnovější „stable“ verzi systému[2] s označením *Debian 7.5.0 wheezy* (vydáno 26. dubna 2014).

Pro ověření důležitosti aktualizace jádra systému, bylo provedeno referenční měření (zátěžové testy se SynFlood útokem) před aktualizací a po ní. Na obrázku 9.1 můžeme vidět srovnání těchto dvou měření. Z obrázku je jasně patrná vyšší odol-



Obr. 9.1: Úspěšnost serveru s novou a starší verzi systému při SynFlood útoku.

nost novější verze systému oproti starší verzi. Zatímco novější verze systému dokáže odolávat SynFlood útoku při hodnotě 1 100 DDoS paketů za sekundu se 100%

[2] ke dni 1. 5. 2014

úspěšností, starší verze systému při stejné hodnotě útoku dosahuje účinnosti pouhých 5 %[3]. Z naměřených výsledků tedy vyplývá výše zmíněná důležitost udržovat systém aktualizovaný.

Za referenční hodnoty pro následné určování účinnosti implementovaných ochran jsou označeny výsledky z měření aktualizovaného systému.

9.3 Ochrana pomocí Apache modulu

Jako speciální Apache modul pro ochranu serveru před DoS a DDoS útoky byl zvolen modul *mod_evasive* [23]. Modul je určen především proti útokům na webové služby (HTTPFlood, útoky hrubou silou, apod.), často se ale také uvádí obecně jako ochrana proti DoS a DDoS útokům. Z toho důvodů byl zařazen i zde jako ochrana proti SynFlood útoku. Na základě výsledků z testování budeme moci jednoznačně říci, zda je, či není tento modul vhodný pro ochranu před SynFlood útokem.

Princip detekce spočívá ve vytvoření vnitřní dynamické hash tabulky IP adres a jednotného identifikátoru zdroje (URI) za účelem dočasné blokace jakékoliv IP adresy, která překročí stanovený počet požadavků za sekundu na danou adresu či webovou stránku [23].

9.3.1 Implementace a konfigurace

Použitím následujících příkazů dojde ke stažení modulu ze stránek autora, dekompresi souboru a jeho instalací:

```
cd /usr/src
wget wget http://www.zdziarski.com/blog/wp-content/uploads/2010/02/
   mod_evasive_1.10.1.tar.gz
tar xzf mod_evasive_1.10.1.tar.gz
cd mod_evasive
apxs2 -cia mod_evasive20.c
```

Do konfiguračního souboru webového serveru `/etc/apache2/apache2.conf` je následně potřeba přidat nastavení instalovaného modulu:

```
<IfModule mod_evasive20.c>
#velikost hash tabulky
DOSHashTableSize    3097
```

[3]Úspěšnost serveru 5 % odpovídá pouze spojením, které proběhly do doby, než začal být DDoS útok aktivní (prvních 32 spojení). Prakticky to znamená, že od chvíle, kdy byl útok aktivní, nedošlo k žádnému úspěšnému spojení se serverem.

```
#počet požadavků na stejnou stránku webu za dobu DOSPageInterval
DOSPageCount        3
#počet požadavků na jakýkoliv objekt webu za dobu DOSSiteInterval
DOSSiteCount        35
#časový interval pro daný počet jednotek DOSPageCount (v sekundách)
DOSPageInterval     1
#časový interval pro daný počet jednotek DOSSiteCount (v sekundách)
DOSSiteInterval     1
#doba dočasné blokace IP adresy (v sekundách)
DOSBlockingPeriod   30
</IfModule>
```

Po restartu webového serveru Apache bude modul aktivní.

9.3.2 Výsledky testování

Ověření funkčnosti samotného modulu bylo provedeno mnohonásobným otevřením webové stránky pomocí webového prohlížeče. Při překročení limitu byla stránka na určitou dobu (v našem případě na 30 sekund) nedostupná.

Úspěšnost serveru po implementaci modulu je ale stejná jako u referenčního měření. Schopnost modulu chránit server před SynFlood útokem je tudíž nulová. Bylo provedeno několik testů s různou konfigurací počtu požadavků za daný interval. Při žádné z konfigurací nebyla účinnost ochrany zaznamenána.

Důvod neúčinnosti ochrany byl zjištěn z analýzy průběhu testování. Z logovacích souborů je zřejmé, že modul vyhodnocuje pouze spojení, která byla úspěšně navázána (tedy prošla regulérním „three-way handshake“ procesem). Samotný SynFlood útok ovšem spočívá v tom, že u jeho zaslaných paketů k úspěšnému navázáni spojení nedojde. Prakticky to znamená, že zatímco je modul připraven blokovat na aplikační vrstvě ISO/OSI modelu příchozí IP adresy, které překračují nastavené limity, samotný útok probíhá o vrstvu níže na transportní vrstvě, kde jsou útokem vyčerpány prostředky pro navazování nových TCP spojení.

Z podrobnější analýzy principu fungování modulu *mod_evasive* na základě jeho testování je tedy zřejmé, že není vhodný pro ochranu před útokem SynFlood.

9.4 Ochrana úpravou proměnných jádra Linuxu

Další nabízenou možností, jak chránit server před SynFlood útokem je úprava a přenastavení hodnot proměnných v jádře operačního systému. Jednotlivých proměnných je v jádře linuxového systému celá řada a jejich hodnoty mají přímý vliv na

funkčnost systému [24]. Jejich změnou lze například navýšit velikost zásobníku určeného k ukládání stavů TCP spojení, čímž může server přijmout více TCP spojení. Dalším příkladem může být úprava hodnot různých časovačů podílejících se na procesu navazování TCP spojení za účelem snížení doby čekání na potvrzení zaslaného požadavku apod.

Vývojáři do jádra linuxového systému implementují také různé ochranné funkce proti DoS a DDoS útokům. Konkrétně proti SynFlood útoku byla v roce 1996 navržena ochrana s názvem *SYN cookies* [25]. Princip ve stručnosti spočívá v tom, že při přijetí SYN paketu serverem je vytvořena tzv. „cookie“, která je výsledkem hash funkce generované z hodnot zdrojové IP adresy, zdrojového portu, sekvenčního čísla a dalších určujících hodnot. Není tedy vytvořen nový stav SYN_RECV, pro který by bylo nutné rezervovat systémové prostředky. Server pak odesílá zpět paket SYN/ACK s „cookie“. Ve chvíli kdy server obdrží odpověď ACK, ověří přijatou „cookie“ a v případě, že je korektní, naváže legitimní spojení. Systémové prostředky jsou tedy rezervovány až na konci „three-way handshake“ procesu a pouze pro ověřený zdroj.

Tyto způsoby ochrany ovšem server před samotným útokem neochrání, pouze mohou zvýšit míru jeho odolnosti vůči útoku. V případě, že útočník svůj útok navýší, ochrana se stane nedostatečnou. Nakolik jsou tato ochranná řešení účinná v praxi, nám ukáži výsledky testování.

9.4.1 Implementace a konfigurace

Implementace příslušných ochranných nastavení je poměrně jednoduchá. Prakticky se jedná o zápis nové hodnoty do proměnné, která přísluší k dané funkci systému. K zápisu hodnoty do proměnné lze použít příkaz `sysctl` (hodnota zůstane v proměnné uložena i po restartu systému) nebo `echo` (po restartu systému se proměnná nastaví na výchozí hodnotu).

Důležité je také upozornit na skutečnost, že ne všechny proměnné je vhodné upravovat. Většina z nich je velmi dobře odladěná a jejich změnou lze narušit stabilitu celého operačního systému [24]. Pro účely testování proto byly vybrány pouze proměnné úzce spjaté s řízením navazování TCP spojení.

```
echo 8192 > /proc/sys/net/ipv4/tcp_max_syn_backlog
#výchozí hodnota: 128
echo 1 > /proc/sys/net/ipv4/tcp_synack_retries
#výchozí hodnota: 3
echo 8192 > /proc/sys/net/core/somaxconn
#výchozí hodnota: 128
```

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

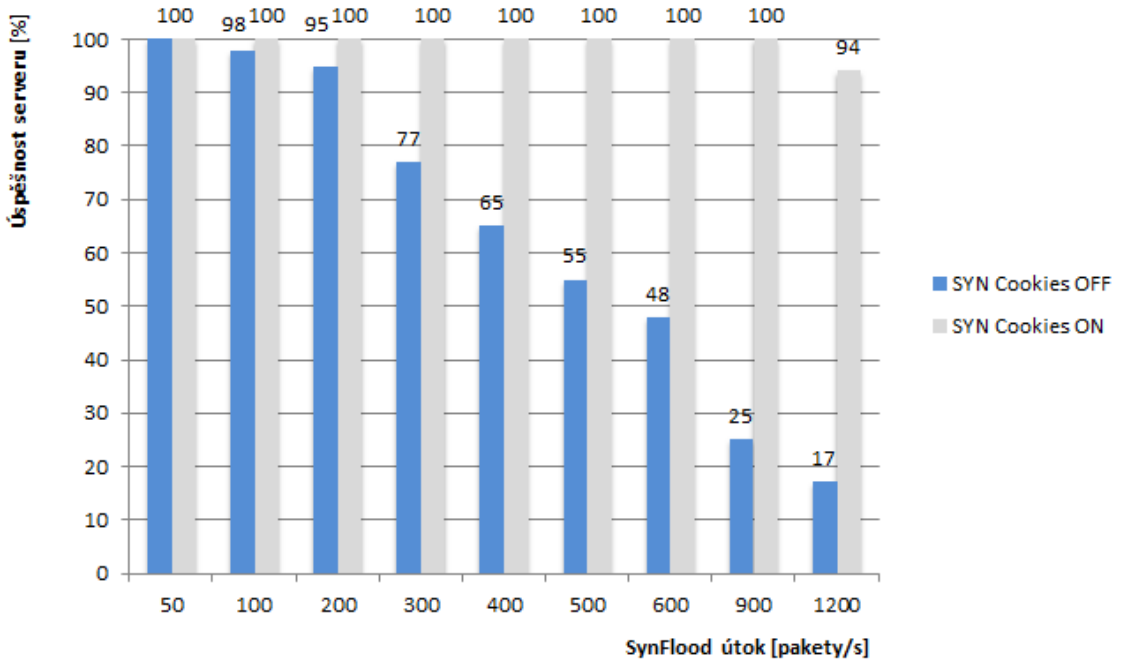
První proměnná `tcp_max_syn_backlog` definuje velikost zásobníku pro ukládáni „napůl“ otevřených spojení, tzn. spojení, kdy byl přijat pouze SYN paket. Snahou je tento zásobník zvětšit, aby se oddálilo jeho přeplnění při útoku. Druhá proměnná `tcp_synack_retries` určuje dobu čekání na přijetí paketu ACK. Čím delší je tato doba, tím déle jsou v zásobníku udržována „napůl“ otevřená spojení. O SynFlood útoku víme, že nikdy žádný ACK paket nezašle, proto je potřeba tuto dobu zkrátit, aby „napůl“ otevřená spojení byla co nejdříve ukončena. Výchozí hodnota 3 znamená, že server čeká na přijetí paketu ACK 45 sekund. Nová hodnota 1 odpovídá době čekání 9 sekund. Další proměnná `somaxconn` určuje maximální počet soketů, které mohou být na serveru otevřeny. Snahou je opět tento počet navýšit.

Poslední proměnná `tcp_syncookies` aktivuje (hodnota 1) a deaktivuje (hodnota 0) ochrannou funkci *SYN cookies*. Zde je potřeba ovšem podotknout, že tato funkce je aktivní již ve výchozím stavu po instalaci operačního systému. Referenční měření prováděné v části 9.2 tudíž tuto ochranu již zahrnuje. Aby bylo možné vyhodnotit také účinnost této ochrany, bude zvlášť proveden test, kde bude zmíněná ochrana deaktivována. Výsledek testu bude následně srovnán s referenčním měřením.

9.4.2 Výsledky testování

Testování bylo prováděno po každé změně hodnoty proměnné, aby bylo možné zjistit, které nastavení mělo na úspěšnost serveru největší vliv. Proveden byl také test, zahrnující nastavení všech výše zmíněných hodnot proměnných. Žádný ze zásahů do proměnných jádra operačního systému ovšem nepřinesl měřitelný rozdíl úspěšnosti serveru oproti referenčnímu testování v části 9.2. Zde je potřeba si uvědomit, že velikost generované zátěže je od 1 000 DDoS paketů/s až po 2 000 DDoS paketů/s. Server čeká 9 sekund na potvrzení spojení než dané spojení zahodí. To znamená, že během této doby by potřeboval „uskladnit“ minimálně 9 000 „napůl“ otevřených spojení. Vzhledem k velikosti generované zátěže se vliv provedené úpravy proměnných neprojeví.

Pro zjištění účinnosti ochrany *SYN cookies* bylo provedeno měření, kdy byla ochrana deaktivována. Srovnání úspěšnosti serveru při zapnuté a vypnuté ochraně je ve formě grafu na obrázku 9.2. Při vypnuté ochranné funkci můžeme pozorovat postupný pokles úspěšnosti serveru v závislosti na zvyšující se generované zátěži SynFlood útoku. Pokles úspěšnosti na 95 % je zaznamenám již při zátěži 200 DDoS paketů/s. Pokud se podíváme na obdobný pokles úspěšnosti serveru (94 %) se zapnutou ochranou, zjistíme, že tomu bylo až u hodnoty zátěže 1 200 DDoS paketů/s. Server s aktivní ochrannou funkcí *SYN cookies* tedy dokáže při zachování stejné



Obr. 9.2: Úspěšnost serveru s aktivní a neaktivní ochranou *SYN cookies*.

úspěšnosti (cca 95 %) odolat zátěži vyšší o 1 000 DDoS paketů/s. Ze získaných výsledků tedy jasně vyplývá, že implementace ochrany *SYN cookies* na serveru má z hlediska ochrany vůči SynFlood útoku své důležité opodstatnění.

9.5 Ochrana pomocí Iptables

Linuxové jádro obsahuje subsystém *Netfilter*, který je zodpovědný za síťový provoz v operačním systému. *Iptables* je pak nástroj na správu tohoto subsystému [26].

Ochrana je tedy založena na principu definování stavových i nestavových pravidel firewallu tak, aby byl útok těmito pravidly zachycen dříve, než naruší vnitřní funkci systému, případně aby byl útok omezen v co největší možné míře. Ochrany v této skupině budou rozděleny do dvou podskupin podle toho, na základě jakého principu bude server před SynFlood útokem chráněn.

9.5.1 První sada pravidel

Do první podskupiny jsou zařazeny pravidla firewallu, která využívají iptables moduly sloužící k nastavení limitů pro příchozí spojení. Použitými moduly jsou *limit*, *connlimit* a *state* spolu s modulem *recent*. Snahou je zachytit pakety generované útokem a propustit legitimní provoz.

Implementace a konfigurace

Výchozí politika firewallu je pro následné testování nastavena na „ACCEPT“, čili pakety nevyhovující žádnému pravidlu budou akceptovány.

U prvního modulu *limit* budou přijatá všechna příchozí spojení, dokud nebude překročen nastavený maximální limit `-limit-burst` za určitý časový okamžik daný hodnotou `-limit`. Implementaci provedeme těmito příkazy:

```
iptables -N syn-flood
iptables -A INPUT -p tcp --syn -j syn-flood
iptables -A syn-flood -m limit --limit 5/s --limit-burst 5 -j RETURN
iptables -A syn-flood -j DROP
```

První pravidlo vytvoří pomocný řetezec, do kterého následující pravidlo odfiltruje příchozí SYN pakety. Třetí pravidlo stanovuje limit pro pakety odfiltrované do pomocného řetezce. Pokud pakety pravidlu vyhoví (tzn. nepřekročí stanovenou hodnotu), jsou vráceny zpět do hlavního řetězce *INPUT*. Pakety které pravidlu nevyhoví (tzn. překročí stanovenou hodnotu) budou zahozeny.

Druhý modul *connlimit* umožňuje omezit počet paralelních spojení z jedné konkrétní IP adresy nebo i z celého IP rozsahu (s pomocí `-connlimit-mask`). Pravidlo zavedeme přímo do hlavního vstupního řetězce *INPUT*:

```
iptables -A INPUT -p tcp --syn -m connlimit --connlimit-above 5
   -j DROP
```

Pakety, které překročí stanovený limit jsou zahozeny.

Poslední, třetí modul *state* se často využívá společně s modulem *recent*. Pomocí modulu *recent* lze omezit počet spojení přicházejících z jedné konkrétní IP adresy za určitou časovou jednotku a pomocí modulu *state* lze rozpoznat k jakému stavu spojení přísluší příchozí paket. V našem případě nás zajímá stav *NEW*, který znamená, že paket začal nové spojení nebo patří ke spojení, které ještě neposílalo pakety v obou směrech. Pravidla opět zavedeme přímo do hlavního vstupního řetězce *INPUT*:

```
iptables -I INPUT -p tcp -m state --state NEW -m recent --set
iptables -I INPUT -p tcp -m state --state NEW -m recent --update
   --seconds 10 --hitcount 5 -j DROP
```

Prvním pravidlem se paket označí a druhým je zkontrolován, zda již nepřekračuje stanovený limit. Pokud paket limit překročí, je zahozen.

Výsledky testování

Na server byla aplikována pravidla jednotlivých modulů, přičemž pro každý modul bylo provedeno několik desítek testů s různou konfigurací počtu požadavků za daný interval. Cílem opakovaných testů bylo pomocí snižování (zvyšování) limitů aplikovaných pravidel najít hranici, při které bude legitimní síťový provoz ještě akceptován a síťový provoz generovaný útokem blokován. S žádným výše popsaným modulem se ale tuto hranici najít nepodařilo. Od určité hranice sice docházelo k blokování útoku, ale stejně tak docházelo i k blokování legitimních spojení. Schopnost modulů chránit server před generovaným DDoS útokem SynFlood byla tedy nulová.

První příčinou neúspěchu je skutečnost, že moduly určené k omezování počtu přijatých paketů (spojení) nedokáži rozeznat, zda jsou příchozí pakety od legitimního uživatele nebo od útočníka. Nelze tedy zvlášť nastavit pravidla pro legitimní a nelegitimní provoz. Druhá příčina neúspěchu tkví v tom, že útok je generován z velkého počtu IP adres, jelikož se jedná o distribuovaný DoS útok. Útok vysílaný z jedné konkrétní IP adresy je tedy zanedbatelný a pravidla pro omezení počtu spojení za jednotku času na něj nebudou reagovat. Platí, že čím více zdrojových IP adres má útočník k dispozici, tím méně častěji se bude opakovat útok z jedné konkrétní IP adresy (při zachování stejného datového toku). V našem případě je vždy dříve omezen provoz legitimního uživatele, než provoz generovaný DDoS útokem.

Aplikaci výše uvedených pravidel lze tedy doporučit pouze k ochraně proti DoS útokům (případně proti malým DDoS útokům), při kterých je zřejmá vysoká aktivita konkrétní zdrojové IP adresy útočníka. Vždy je ale potřeba dbát na to, aby nastavené limity neomezily i legitimní provoz.

9.5.2 Druhá sada pravidel

Druhá podskupina reprezentuje nastavení firewallu, které propustí síťovou komunikaci pouze z ověřených zdrojů a pouze na porty poskytovaných služeb. Zbylý příchozí síťový provoz je zahazován. Fungovat tedy budou pouze provozované služby serveru a to pouze pro ověřené IP adresy. Aplikace takovéto sady pravidel je v laboratorních podmínkách poměrně jednoduchá, protože známe rozsahy IP adres, ze kterých je generován legitimní provoz. V praxi je tato situace o mnoho těžší. V případě, že se jedná o server poskytující své služby do internetové sítě, nelze předem určit, z jaké zdrojové adresy legitimní požadavek přijde. Cílem měření tedy není otestovat konkrétní sadu iptables pravidel, ale změřit velikost zátěže, kterou je server schopen s pomocí iptables odfiltrovat. Zjistíme tak, jak silný musí útok být, aby dokázal „překonat“ pravidla firewallu.

Implementace a konfigurace

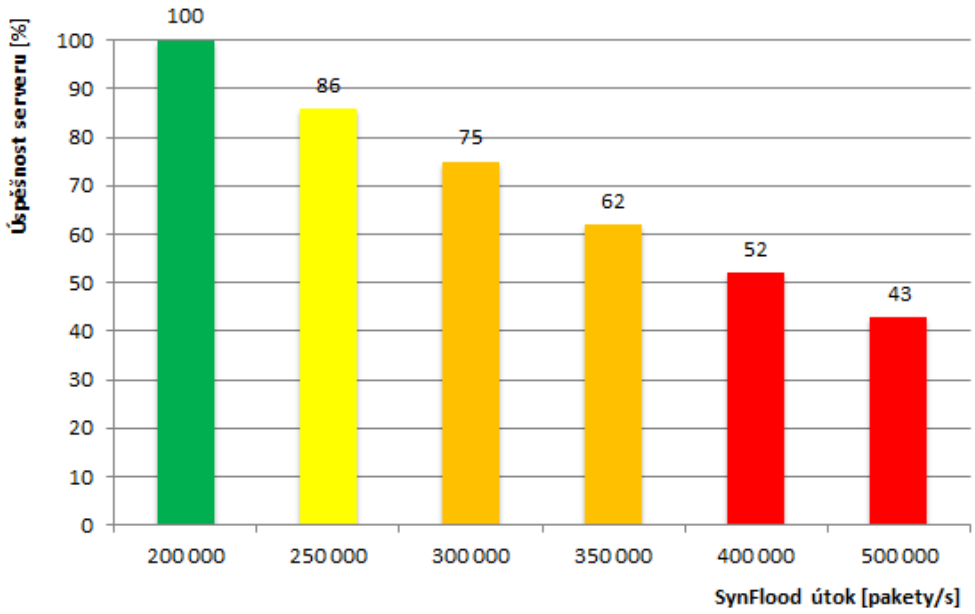
Pro změření velikosti zátěže, kterou je server schopen s pomocí iptables odfiltrovat, nám v laboratorních podmínkách postačí definovat pár jednoduchých příkazů:

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

iptables -A INPUT -i eth1 -s 10.0.0.0/16 -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -i eth1 -d 10.0.0.0/16 -p tcp -j ACCEPT

iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -i eth0 -p tcp --sport 22 -j ACCEPT
```

První tři příkazy nastaví restriktivní bezpečnostní politiku, čili co není přímo povoleno, je zakázáno. Následující dva příkazy povolí legitimní provoz generovaný zařízením *Avalanche* v příchozím i odchozím směru. Poslední dva pravidla povolí vzdálené připojení pomocí SSH.

Výsledky testování

Výsledný graf úspěšnosti serveru v závislosti na generované zátěži SynFlood útoku je na obrázku 9.3. Implementovaná pravidla firewallu pomohla serveru odolávat útoku



Obr. 9.3: Úspěšnost serveru při filtrování útoku pomocí iptables pravidel.

až do velikosti zátěže 200 000 DDoS paketů/s. Nad touto hranicí již firewall nedokáže nežádoucí pakety spolehlivě filtrovat a zahazovat. S poloviční úspěšnosti dokáže server pracovat ještě při zátěži 400 000 DDoS paketů/s.

9.6 Shrnutí a doporučení

Cílem této kapitoly bylo otestovat různé typy „end-host“ ochran proti SynFlood útoku, zjistit nakolik jsou účinné a určit, zda je má smysl v praxi na koncovém zařízení implementovat.

Velice zajímavých výsledků bylo dosaženo již v úvodním referenčním měření, kde byla testována starší verze jádra linuxového systému a verze novější proti SynFlood útoku. Ve výsledném srovnání (viz obr. 9.1) byla velice zřetelná vyšší odolnost novější verze jádra systému oproti verzi starší. Bylo tak prakticky ověřeno, že udržovat systém aktuální je z hlediska odolnosti vůči útokům velmi důležité.

První testovanou ochranou byl speciální modul *mod_evasive* webového serveru Apache. Z analýzy výsledků ovšem vyplynulo, že modul není vhodný pro ochranu před SynFlood útokem.

Následně byla testována ochrana, která spočívala v navýšení (snížení) hodnot proměnných, jenž jsou spjatá s navazováním TCP spojení. Žádný ze zásahů do proměnných jádra operačního systému ovšem nepřinesl měřitelný rozdíl úspěšnosti serveru oproti referenčnímu testování. Otestována byla také funkce *SYN cookies*, která pro změnu přinesla velmi pozitivní výsledky (viz obr. 9.2). Její aktivace zajistí serveru mnohem vyšší odolnost vůči SynFlood útoku.

Ochrana pomocí Iptables byla rozdělena na dvě části. V první části byly testovány iptables moduly sloužící k nastavení limitů pro příchozí spojení. Slabinou těchto modulů je fakt, že nedokáži rozpoznat legitimní provoz od provozu DDoS útoku. Často se tak může stát, že nastavený limit omezí nejen DDoS útok, ale také legitimní provoz. Tato pravidla lze tedy doporučit pouze pro ochranu před DoS útoky (případně proti velmi malým DDoS útokům), při kterých je zřejmá vysoká aktivita konkrétní zdrojové IP adresy útočníka.

Poslední měření patří do druhé části ochran pomocí Iptables. Cílem bylo změřit velikost zátěže, kterou je server schopen s pomocí iptables pravidel odfiltrovat. Naměřená byla hodnota 200 000 DDoS paketů/s. Při této velikosti zátěže dokáže firewall filtrovat a zahazovat nežádoucí pakety se 100% úspěšnosti.

Efektivně nastavená pravidla firewallu jsou tedy nejúčinnější „end-host“ ochranou. Pro efektivní ochranu pomocí Iptables lze ve stručnosti doporučit restriktivní bezpečnostní politiku a povolení pouze přístupu k poskytovaným službám. Dále preventivní blokování IP adres, které jsou známy svou DDoS aktivitou (seznam infi-

kovaných adres poskytuje například zdroj [27]). Neméně důležitý je pak monitoring stavu serveru a rychlá reakce na případný útok. Stále ale platí, že pokud bude útok směřován na služby poskytované serverem, jedinou ochranou je aktualizovaný systém a aktivovaná ochrana *SYN cookies*. Při takovémto útoku klesá odolnost testovaného serveru z 200 000 DDoS paketů/s na pouhých 1 100 DDoS paketů/s.

10 ZÁVĚR

Jednotlivé cíle bakalářské práce byly realizovány v rámci teoretické a praktické části. Úvod teoretické části byl věnován operačnímu systému Linux, jeho historickému vývoji a výběru distribuce Debian. Následovalo seznámení s aplikacemi Apache a Vsftpd, které poskytují služby HTTP a FTP serveru. V další části byla popsána problematika testování zátěže. Bylo zde představeno klíčové zařízení pro realizaci zátěžových testů, Spirent Avalanche 3100. Důležitou součástí této kapitoly byl popis aplikací *TestCenter Layer 4-7 Application* a *TestCenter Results Analyzer* sloužící k práci se zařízením a k interpretaci výsledků z testování. Cílem popisu specifik aplikací bylo získat potřebné znalosti pro praktickou realizaci zátěžových testů. Následující kapitola se věnovala DDoS útokům. Byla zde provedena analýza současného stavu v této oblasti a stručný rozbor jednotlivých DDoS útoků, které poskytuje zařízení Spirent Avalanche 3100. Dále bylo ukázáno obecné řešení, jak postupovat při návrhu ochranných prostředků proti DDOS útokům.

V praktické části byl na server nejdříve nainstalován operační systém Linux v distribuci Debian. Poté byly implementovány a nakonfigurovány webové a datové služby poskytované aplikacemi Apache a Vsftpd. Instalován byl také firewal FireHOL a vzdálený přístup na server OpenSSH.

Následovala realizace zátěžových testů (bez DDoS útoků), při které bylo využito prostředků virtualizační platformy VMware vSphere. Účelem zátěžových testů bylo zjistit, při jaké zátěži bude server dosahovat stanovených úspěšností, přičemž testování bylo prováděno pro webové a datové služby. Hlavním cílem této praktické časti bylo poukázat na metodiku měření a vyhodnocování získaných výsledků.

Další část se již věnovala zátěžovým testům s DDoS útoky. Jako testovaný server sloužilo zařízení firmy Hewlett-Packard vybavené dvěmi síťovými kartami s teoretickou přenosovou rychlostí 1Gbit/s. Měřen byl vliv DDoS útoků na webové služby serveru. Konkrétně byly realizovány útoky SynFlood, UDPFlood, XMasTree, ResetFlood a ARPFlood. Na základě získaných výsledků byl jako nejúčinnější útok zvolen SynFlood, kterému server dokázal se 100% úspěšnosti odolávat pouze do hodnoty zátěže 825 DDoS paketů/s. Téměř stejných výsledků dosáhl i útok UDPFlood. Oproti tomu u útoku ARPFlood nebyl zaznamenán žádný vliv na úspěšnost serveru.

Poslední praktická část byla věnována testování ochranných prostředků proti SynFlood útoku, které byly postupně implementovány na testovaný server. Shrnutí této části spolu s doporučením je uvedeno v závěru kapitoly (9.6). Z výsledků provedených testů vyplynula důležitost aktualizace jádra operačního systému, aktivace ochranného mechanizmu *SYN cookies* a především důležitost efektivně nastavených pravidel firewallu, který dokáže odolávat zátěži do 200 000 DDoS paketů/s.

LITERATURA

[1] *The Apache Software Foundation* [online]. 2012 [cit. 2013-12-10]. Dostupné z URL: <http://www.apache.org/>.

[2] HUNT, Craig. *Linux: síťové servery.* Praha: SoftPress, c2003, 672 s. ISBN 80-864-9759-3.

[3] KYSELA, Martin. *Přecházíme na Linux.* 1. vyd. Brno: Computer Press, 2003, 191 s. ISBN 80-722-6844-9.

[4] SOBELL, Mark G. *Linux: praktický průvodce.* 1. vyd. Praha: Computer Press, 1999, 946 s. ISBN 80-722-6190-8.

[5] SIEVER, Ellen. *LINUX v kostce.* 1. vyd. Praha: Computer Press, 1999, 560 s. ISBN 80-722-6227-0.

[6] SHAH, Steve. *Administrace systému Linux: jak porozumět svému počítači : podrobný průvodce začínajícího administrátora.* 2. vyd. Praha: Grada, 2003, 553 s. ISBN 80-247-0641-5.

[7] Debian. *Wikipedie, otevřená encyklopedie* [online]. 2010, 2013-10-14 [cit. 2013-12-10]. Dostupné z URL: <http://cs.wikipedia.org/wiki/Debian>.

[8] SILVA, Gustavo N. APT HOWTO (Obsolete Documentation). SOFTWARE IN THE PUBLIC INTEREST, Inc. *Debian* [online]. 2004 [cit. 2013-12-10]. Dostupné z URL: <http://www.debian.org/doc/manuals/apt-howto/index.cs.html#contents>.

[9] June 2013 Web Server Survey. *Netcraft* [online]. 2013 [cit. 2013-12-10]. Dostupné z URL: <http://news.netcraft.com/archives/2013/06/06/june-2013-web-server-survey-3.html>.

[10] FAQ Httpd Wiki. *The Apache Software Foundation Projects* [online]. 2013 [cit. 2013-12-10]. Dostupné z URL: <http://wiki.apache.org/httpd/FAQ#Is_there_any_more_information_available_on_Apache_httpd.3F>.

[11] EVANS, Chris. *Vsftpd* [online]. 2011 [cit. 2013-12-10]. Dostupné z URL: <https://security.appspot.com/vsftpd.html>.

[12] SCAMBRAY, Joel, George KURTZ a Stuart MCCLURE. *Hacking bez tajemství.* 2., aktualiz. vyd. Praha: Computer Press, 2002, xxviii, 625 s. Komunikace. ISBN 80-722-6644-6.

[13] HATCH, Brian. *Hacking bez tajemství LINUX*. Vyd. 1. Brno: Computer Press, 2003, 644 s. ISBN 80-722-6869-4.

[14] Global Application & Network Security Report 2013. *Radware, Ltd.* [online]. 2014 [cit. 2014-04-28]. Dostupné z URL: <http://www.atsweb.it/fileadmin/ATSWeb/downloads/Radware_2013_ERT_Report.pdf>.

[15] BAO, Xuhua; HONG, Hai. NSFOCUS DDoS Threat Report 2013. *NSFOCUS Information Technology Co., Ltd.* [online]. 2014 [cit. 2014-04-28]. Dostupné z URL: <http://en.nsfocus.com/SecurityReport/NSFOCUSDDoSThreatReport2013.pdf>.

[16] Prolexic Quarterly Global DDoS Attack Report Q4 2013. *Prolexic Technologies., Inc* [online]. 2014 [cit. 2014-04-28]. Dostupné z URL: <http://www.akamai.com/dl/akamai/prolexic-q42013-global-attack-report-us.pdf>.

[17] SEJK, Vratislav. Co jste mohli udělat proti (D)Dos útokům? *Unicorn Systems* [online]. 2013 [cit. 2013-12-10]. Dostupné z URL: http://www.unicornsystems.eu/cz/novinky/clanek/co-jste-mohli-udelat-proti-ddos-utokum.html.

[18] THE INTERNET ENGINEERING TASK FORCE. *RFC 4987* [online]. 2007 [cit. 2013-12-10]. Dostupné z URL: <http://tools.ietf.org/html/rfc4987>.

[19] SEDLÁK, Jan. NIX.CZ pracuje na ochraně před masivními DDoS útoky. *Živě.cz* [online]. 2013 [cit. 2013-12-10]. Dostupné z URL: http://connect.zive.cz/clanky/nixcz-pracuje-na-ochrane-pred-masivnimi-ddos-utoky/sc-320-a-170617.

[20] Doporučení pro případ napadení DDoS útokem. *Národní centrum kybernetické bezpečnosti* [online]. 2013 [cit. 2013-12-10]. Dostupné z URL: http://www.govcert.cz/cs/informacni-servis/aktuality/doporuceni-pro-pripad-napadeni-ddos-utokem---jak-se-zachovat-a-jak-postupovat/.

[21] Virtualizace serverů produkty VMWARE vSphere. *VAHAL s.r.o* [online]. 2009 [cit. 2013-12-10]. Dostupné z URL: <http://www.vahal.cz/cz/produkty/software/vmware-virtualizace.html>.

[22] VLAN Security White Paper. *Cisco* [online]. [cit. 2014-05-18]. Dostupné z URL: <http://www.cisco.com/en/US/products/hw/switches/ps708/products_white_paper09186a008013159f.shtml#wp39054>.

[23] ZDZIARSKI, Jonathan. mod_evasive. *Jonathan Zdziarski's Domain* [online]. [cit. 201-05-10]. Dostupné z URL: <http://www.zdziarski.com/blog/?page_id=442>.

[24] ANDREASSON, Oskar. Ipsysctl tutorial 1.0.4 *Frozentux.net* [online]. 2002 [cit. 2014-05-16]. Dostupné z URL: <https://www.frozentux.net/ipsysctl-tutorial/ipsysctl-tutorial.html#AEN485>.

[25] BERNSTEIN, Daniel J. SYN cookies. *cr.yp.to* [online]. [cit. 2014-05-16]. Dostupné z URL: <http://cr.yp.to/syncookies.html>.

[26] Iptables. *Ubuntu.cz* [online]. 2012 [cit. 2014-05-16]. Dostupné z URL: <http://wiki.ubuntu.cz/bezpečnost/firewall/iptables>.

[27] Top 10 Reports. *Internet Storm Center* [online]. [cit. 2014-05-18]. Dostupné z URL: <https://isc.sans.edu//top10.html>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

APT	Advanced Packaging Tool
ARP	Address Resolution Protocol
CERT	Computer Emergency Response Team
CIRT	Critical Incident Response Team
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
FTP	File Transfer Protocol
GPL	General Public License
GUI	Graphic User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDPS	Intrusion Detection Prevention System
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IPS	Intrusion Prevention System
ISO	International Standards Organization
OSI	Open Systems Interconnection
LUI	Line User Interface
MAC	Media Access Control
MSN	Microsoft Network
NCKB	Národní centrum kybernetické bezpečnosti

NIC	Network Interface Controller
PHP	Hypertext Preprocessor
POP	Post Office Protocol
QoE	Quality of Experience
QoS	Quality of Service
RFC	Request for Comments
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URI	Uniform Resource Identifier

SEZNAM PŘÍLOH

A OBSAH PŘILOŽENÉHO CD

Na přiloženém CD je uložena elektronická podoba (.pdf) této bakalářské práce.